FINAL SYNOPSIS OF THESIS ENTITLED

Development of Subjective Examination Assessment Based on Arabic Automated Question Answering System

Submitted to

Dr. Babasaheb Ambedkar Marathwada University, Aurangabad

For the award of

Doctor of Philosophy in Computer Science

(Under Faculty of Science)

By

Samah Ali Ali Saleh Alazani
(Research Student)
Department of Computer Science and Information Technology,
Dr. Babasaheb Ambedkar Marathwada University, Aurangabad

Under the guidance of

Dr. C. Namrata Mahender
Assistant Professor,
Department of Computer Science and Information Technology,
Dr. Babasaheb Ambedkar Marathwada University,
Aurangabad – 431 004 (MS) India.

INDEX

1. **INTRODUCTION ……. 1**

   1.1 Challenges of Arabic QAS ……..…… 1

   1.2 Different types of Question-Based on Blooms Taxonomy …...……. 2

2. **LITERATURE REVIEW ……. 3**

   2.1 A Survey on QAS ..…….. 4

   2.2 A Survey on Deep Learning Arabic QAS ……..… 5

3. **MOTIVATION AND OBJECTIVES ……..…… 6**

4. **PROPOSED METHODOLOGY ……..…….……...** 7

5. **PRACTICAL EXPERIMENT…….……. 9**

   5.1 Experiment one…… 11

   Data collection ……11

   5.1.2 Pre-Processing Stage …….…... .12

   5.1.3 Processing Stage ……13

   5.1.4 Algorithm of the proposed Hamming model ……..… 14

   5.2 Experiment two…….19

   5.2.1 Proposed Model …….20

   7 Experiment three ……...20

   7.1 Prepare Dataset …….…... .21

   7.2 Collecting Data …….…..... 21

   7.3 Data Preprocessing …….…..…22

   7.4 Building the Models …….…... 22

8. Result ….....…… 23

8.1 Experiment one..…….…..…….23

8.2 Experiment Two..……..……..…..…25

8.3 Experiment Three..…….……..…….... 34

6. CONCLUSION & FUTURE WORK..…….……43

7. Outline Of Thesis…….…….…….44

8. REFERENCES…….……..…..47

9. PUBLICATION…….…..... 50

GLOSSARY

LIST OF FIGURES

SR. NO.	FIGURE NO.	TITLE	PAGE NO.
1	Figure 1	Proposed Model	7
2	Figure 2	Hamming Distance	8
3	Figure 3	Standard Hamming Distance	9
4	Figure 4	Proposed Edit Distance.	10
		Hierarchical Question Answering System (HQAS).	14
5	Figure 5	m-ary tree with eight nodes with three depths	12
6	Figure 6	types of complete and incomplete multiple words short answers	14
7	Figure7	Hamming-based similarity between TQAS and HQAS in dual-ordered incomplete multiple word short answer.	16
8	Figure 8	Character similarity based on Hamming, Levenshtein, and Jaro_Winkler algorithms between TQAS and HQAS in intra-ordered complete multiple word short answer.	16
9	Figure 9	m-ary tree with 16 nodes with 5 depths.	17
10	Figure 10	deep learning model	19

11	Figure 11	flowchart deep learning models	20
12	Figure 12	distance of the standard Hamming model	22
13	Figure 13	distance of the proposed edit based model	22
14	Figure 14	Difference in the hidden acknowledge assessments on both TQAS and HQAS using Hamming algorithm	24
15	Figure 15	Difference in the hidden acknowledge assessments on both TQAS and HQAS using Levenshtein algorithm.	25
16	Figure 16	Difference in the hidden acknowledge assessments on both TQAS and HQAS using Jaro_Winkler algorithm.	27
17	Figure 17	Distribution of acknowledge performance on different depths of the topic	32
18	Figure 18	Measures of accuracy and val _ accuracy on the training set and the validation set on 10epochs.	32
19	Figure 19	Measures of Loss and val _ loss on the training set and the validation set on 10epochs.	33
20	Figure 20	Measures of accuracy and val _ accuracy on the training set and the validation set on 20 epochs.	33
21	Figure 21	Measures of Loss and val _ loss on the training set and the validation set on 20 epochs.	33
22	Figure 22	Measures of accuracy and val _ accuracy on the training set and the validation set on 50 epochs.	33
23	Figure 23	Measures of Loss and val _ loss on the training set and the validation set on 50 epochs.	35
24	Figure 24	Measures of lstm accuracy and val _ accuracy on the training set and the validation set on 10epochs.	35
25	Figure 25	Measures of lstm loss and val _ loss on the training set and the validation set on 10epochs.	35
26	Figure 26	Measures of accuracy and val _ accuracy on the training set and the validation set on 20 epochs.	36
27	Figure 27	Measures of Loss and val _ loss on the training set and the validation set on 20 epochs.	36
28	Figure 28	Measures of accuracy and val _ accuracy on the training set and the validation set on 50 epochs.	36
29	Figure 29	Measures of Loss and val _ loss on the training set and the validation set on 50 epochs	36
30	Figure 30	Measures of GRU accuracy and val _ accuracy on the training set and the validation set on 10epochs.	38
31	Figure 31	Measures of GRU loss and val _ loss on the training set and the validation set on 10epochs.	38
32	Figure 32	Measures of accuracy and val _ accuracy on the training set and the validation set on 20 epochs.	38

33	Figure 33	Measures of Loss and val _ loss on the training set and the validation set on 20epochs	38
34	Figure 34	Measures of accuracy and val _ accuracy on the training set and the validation set on 50 epochs.	39
35	Figure 35	Measures of Loss and val _ loss on the training set and the validation set on 50 epochs	39
36	Figure 36	RRN – LSTM - GRU Accuracy Results	39
37	Figure 37	RRN – LSTM - GRU Loss Results .	43

LIST OF TABLES

SR. NO.	FIGURE NO.	TITLE	PAGE NO.
1	Table 1	Literature Survey on Question Answering Systems	3
2	Table 2	Comparison between standard Hamming method and proposed Hamming method.	22
3	Table 3	character-based similarity and acknowledge assessment in both TQAS and HQAS using Hamming algorithm	23
4	Table 4	character-based similarity and acknowledge assessment in both TQAS and HQAS using Levenshtein algorithm	24
5	Table 5	character-based similarity and acknowledge assessment in both TQAS and HQAS using Jaro_Winkler algorithm	26
6	Table 6	The actual assessment for the acknowledge about the answer	29
7	Table 7	Difference in accuracy, precision, recall and F1-score in both TQAS and HQAS using Hamming, Levenshtein and Jaro_Winkler algorithms	30
8	Table 8	Performance metrics on RRN model	31
9	Table 9	Performance metrics on LSTM model	34
10	Table 10	Performance metrics on GRU model	36

Acknowledgement

I wish to express my deep sense of gratitude and indebtedness to my guru andresearch guide Dr. C. Namrata Mahender, for the meticulous guidance, valuable discussionand idealistic suggestions during study. She has supported, helped and encouraged me inall ways. I be in awe of the principals, practices and lucky to have company with her always.I am here today, is due to her kindness, cooperation, guidance and support.

I am also grateful to Dr. S. N. Deshmukh, Professor and Head, Department ofComputer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad for his encouragement and administrative help time to time. I also thankful to Dr. R. R. Deshmukh valuable support and encouragement me.

Also thanks to Computational and Psycholinguistic Research Lab Facility supporting this work and Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India**.**

I also thanks to my country and my college to provide me economical support and timely help for completing all process .

Abstract

Automatic subjective assessment based on question answering system represents a desirable trend to assess short and long answers written by the students. Most of the challenges we face When evaluating students answers are how to improve the evaluation of the student answer based on using an Arabic language. In this research, we proposed Hamming model tried to solve the drawbacks of the standard Hamming model by applying the stemming approach to achieve derivative lexical similarity and applying the space padding to deal with unequal lengths of the texts. Also, we have implemented a proposed edit-based Hierarchical question answering system (HQAS) using a traversing by Breadth-First Search (BFS) within an m-ary tree to consider the ignored significant knowledge due to the misspelling at the middle of the dual ordered incomplete answer, the misspelling at middle and the end of the intra-ordered incomplete answer, and the misspelling due to switching in words of the intra-ordered complete answer. It can differentiate among the students based on their significant hidden knowledge and show a distribution of knowledge content on different depths of the topic to determine which of the topic depths the student has the most significant knowledge. And is related to the long answers such as sentences and essay, it introduces a novel way for grading using simple RRN model and advanced RNN models (LSTM and GRU).

1. Introduction

Question answering systems have evolved into a whole new domain. Question answering systems are of emense importance in the present scenario as well as for many purposes such as human-like conversation, machine translation, and summarization. Natural language processing (NLP) is an interdisciplinary field, as its subfield of linguistics, artificial intelligence, knowledge engineering which is generally concerned with the interaction between computer and human, and even mimic human speech and writing, etc. in robots. Natural language processing evolved with the concept of having a thinking machine, which can have interaction as intimated human with humans [1]. Various applications of NLP are very important for the flawless working of machines, like automated translation, speech processing, language understanding, common sense reasoning, automated summarization, automated question answering, and many more [2]. Question answering system in today's scenario of covid-19 becomes more essential, as all over the globe education is provided through online media and examination is one of the important parts of the education system. In the present scenario, objective based (MCQ) question answering is available but there is a huge demand for subjective based online examination system. As many courses require developing and checking the skill and understanding about a concept by a candidate or student a subjective examination is a must. In the question answering system, question processing, answering processing, question generation, answer extraction, question design, and classification and evaluation is a very difficult task. [3]. For the generation of an online system, it has to have a question bank which makes things handy. But the development of automated one is one of the major issues of the question answering system (QAS).

1.1 Challenges of Arabic QAS

Arabic is a Semantic language [4]. It has a population of over 422 million people worldwide. The first language of the Arab and the official language of the United Nations [5]. It is the third most important international language after English and French. The Arabic language has a very rich combination of special features that the computer is difficult to perform [6]. This advantage has created many challenges that researchers have to deal with differently. This section looks at many of these challenges:

1. No capital letters.
2. Lack of linguistic resources.
3. Optional short vowels.
4. Free order sentences.
5. Arabic an inflectional language (word = root +affixes (prefix , infix , suffix).

1.2 Different Types of Questions based on Bloom's Taxonomy

Bloom's taxonomy is a multi-faceted model that promotes progressive learning with six levels of complexity. These six levels include Remembering, Understanding, Applying, Analyzing, Evaluating, and Creating. Bloom's taxonomy encourages students to engage in knowledge at a deeper and more interactive level, to apply what they learn with a real-world perspective, rather than simply taking the information to the board. Questions a frequently used question expression experimental knowledge [7]. A classification system in Bloom's taxonomy is used to describe and classify different levels of human understanding. Teachers generally use Bloom's texts to inform or guide the development of exams (exams and other student learning assessments), curriculum (units, courses, projects, and other learning activities), and teaching methods such as questioning techniques [8]. According to the general agreement, QA Systems are categorized based on the following support purpose:

Open-domain QA: This type of QA handles questions almost anything, and may only depend on public ontologies and world knowledge. On the other hand, these programs often have more available information to them to issue a response. The system takes the NL query as input more than as a collection of words [9].

Closed-domain QA: This type of QA handles questions under certain conditions background (for example, Education or Bank), and is visible as an easy task because NLP programs can exploit particular domain information that is regularly organized into ontologies. Alternatively, a

closed domain may refer to a situation where only a limited number of questions are allowed, as descriptive questions asking rather than process details [10].

2. Literature Survey

2.1 A survey on Question Answering Systems

In this section we have summarized our study on various Question answering systems available in different Languages, which is briefed in Table (1).

Table 1: Literature Survey on Question Answering Systems

No	Author and year of publication	Description / Methodology	Limitations/Future work	Domain	language
1	Asmita Dhokrat , Gite Hanumant R And C.Namrata Mahender, (2012)[11].	Assessment descriptive question based on the tutorial developed. Matching answers used Python with NLTK toolkit for. Synonymy paraphrasing used (Word Net).	The researcher in this paper in future seek to add multiline answers, figures (diagrams), examples, abbreviations, and more contextual different references.	Closed Domain	English

2	Naveed Azim Imran Naqvi Dr. Kashif ur Rehman (2009)[12].	Using SPSS 15.0 version data sheet. Select 100 private companies. the study using ordinal and nominal scales.	In This paper not mention the online subjective exam can be evaluated only looked at the performance certified professional given subjective examination.	Closed Domain	English
3	Kepei Zhang and Jieyu Zhao (2010)[13] .	The methodology in this paper is implement the question classification experiments Support Vector Machines (SVM). the answer extraction is converted into a clustering problem.	In this paper There are some restrictions the named entity tagging is usually limited. If there is a newly named entity, the system cannot recognize it. the classification accuracy is not high enough for the entity category. the answer is not brief enough and the accuracy is low for descriptive questions and system is used in limited areas	Open Domain	Chines
4	Volery, T., et al (2000)[14].	Researched the critical success factor in online education system. This paper suggests to in student expressing knowledge subjective manner during course work.	In this paper only there are two limitations to the study: The first was the small sample size of 47 students. The second that student grades were not available at the time of the data collection.	Closed Domain	English
5	Hyo-Jung Oh et al. (2005)[15].	Define 10 descriptive answer types (DAT) for question. The researcher proposed anew descriptive QA model and presented result of a system.	The system poor for inverted retrieval. System depends on initial patterns and Problem initial patterns influence on retrieval results.	Open Domain	Korean
6	Waheeb Ahmed et al. (2017) [16].	This study presents a supervised support vector machine (SVM) classifier for question classification and answer selection. The performance of	in this paper mentioned for future research, more training data should be added to increase the accuracy of the system.	Closed Domain	Arabic

		answer selection task using the Mean Reciprocal Rank(MRR).			
7	Garima Nanda et al. (2016) [17].	The paper discusses the implementation of a Hindi Language QA system developed using Machine Learning approach. Used the Naïve Baye's classifier. This paper reflects the concepts of similarity and classification provide better results.	Researchers mention as future work the system can be in different languages and can use different classification techniques along with different datasets.	Closed Domain	Hindi

2.2 A survey on Deep learning Arabic Question Answering Systems

The progress on question answering system development in deep learning used a various approaches with nlp and different algorithm techniques. On this work proposed model requires and demand done with twenty tasks of babI dataset of Facebook. The main concept used in QAS is the turning exam, which provides answers based on question patterns found using Artificial Intelligence and Machine Learning techniques. This technique is followed by search engines, natural language question-answering systems, chat bots, and ECA, and comparative studies are conducted for advancement. The ALICE foundation is a widely used standard for developing chat bots with artificial intelligence support. ALICE developed the Artificial Intelligence Markup Language (AIML) software, which includes a scripting language for creating chatbots.The basic tags of AIML are category, which is used to store knowledge bases, pattern, which is used to handle user requests, and template, which is used to handle system responses to the user .

Antoun, Baly & Hajj 2021created the AraELECTRA pre-training text discriminators for Arabic language comprehension. The BERT model's architecture and layering are the same as those of the discriminator network. They adjusted the whole model using the additional layer on reading comprehension tasks in order to fine-tune their method by adding a linear classification layer on top of ELECTRA's output. Numerous Arabic NLP tasks, including reading comprehension, were used to test the model. QA systems in the Arabic language are developing relatively slowly when

compared to English-language QA. This is because there aren't enough NLP tools and datasets for Arabic QA. The sparse technique is used in Arabic OpenQA study for passage retrieval[18].

Almiman et al 2020 highlighted the difficulty of addressing questions in the Arabic community. They investigated the impact of preprocessing and used a variety of similarity features. Through the use of the semantic and lexical similarity features, they created a novel deep neural network ensemble model. The BERT model, which has recently made breakthroughs, was used in the model. The MRR value for the model was 68.86%[19].

Karpukhin et al., 2020 focuses on merging the BERT pre-trained model and a dual-encoder architecture to produce the right dense embedding model using only pairs of questions and answers. Their dense passage retriever creates an index for each passage to be retrieved and employs a dense encoder to transform any text into a dimensional real-valued vector. On a variety of QA datasets, including SQuAD, Natural Questions, and TriviaQA, their suggested model outperformed numerous open-domain QA methods[20].

Guu et al. 2020 a successful method that blends a pre-trained language model with a learnt textual neural knowledge retriever.By requiring the model to select which knowledge to extract and use during inference, this technique explicitly exposes the importance of world knowledge in contrast to models that store knowledge in their parameters[21].

Huang et al., 2020 with help the Arabic OpenQA system run more efficiently. They are used a two-stage (Retriever-Reader) design to build OpenQA systems because it is the most effective and fruitful approach [22].

3. Motivation and Objectives

A. Motivation

Question Answering Systems present a good solution for querying unstructured and structured information. This is why a large scope for development of question answering systems and much work is in progress in various languages. But we observed during our study that only preliminary research work is seen in Arabic language. We found great research scope in this research area, thus we are working on question answering system.

B. Objectives

Major objectives of this research work are:

1) To create content based system for answering the given question.
2) To create an appropriate scheme of answering assessing the answers based on the content of questions.
3) To evaluate candidate single word short answers based on derivate lexical similarity.
4) To Consider an Ignored Knowledge in Misspelled Multiple Words Short Answers.
5) To evaluate long answers using simple and advanced recurrent neural network.

4. **Proposed Methodology**

The proposed research composes of three works. The first work is related to the single word short answers. It introduces a proposed Hamming model to solve the drawbacks of the standard Hamming model by applying the stemming approach to achieve derivative lexical similarity and applying the space padding to deal with unequal lengths of the texts. The second work is related to multiple word short answers, it implements a proposed edit-based Hierarchical question answering system (HQAS) using a traversing by Breadth-First Search (BFS) within an m-ary tree to consider the ignored significant knowledge due to the misspelling at the middle of the dual-ordered incomplete answer, the misspelling at middle and the end of the intra-ordered incomplete answer, and the misspelling due to switching in words of the intra-ordered complete answer. It can differentiate among the students based on their significant hidden knowledge and show a distribution of knowledge content on different depths of the topic to determine which of the topic depths the student has the most significant knowledge. The third work is related to the long answers such as sentences and essays, it introduces a novel way for grading using simple RRN model and advanced RNN models such as LSTM and GRU.

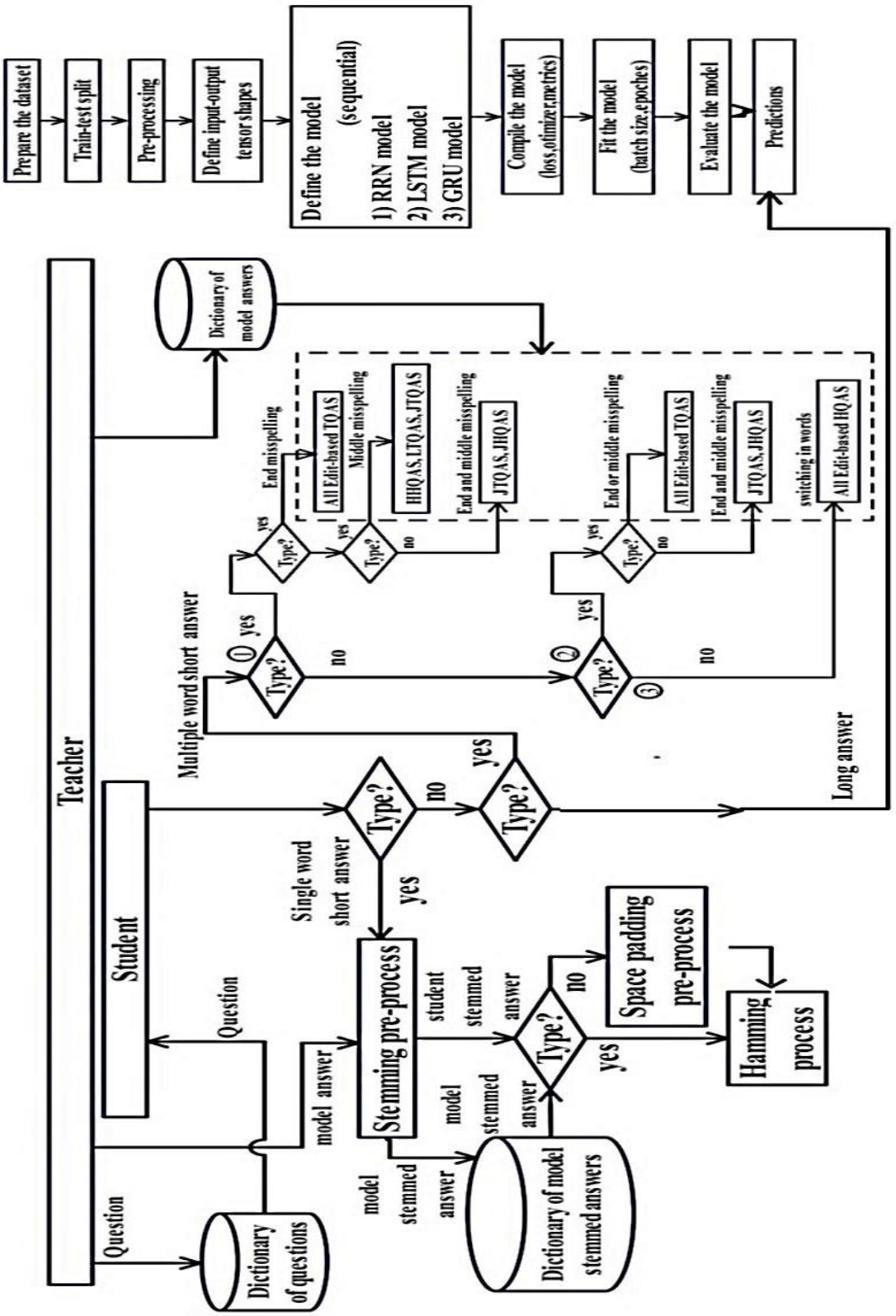


Fig (1). Proposed Model.

5. Practical Experiment

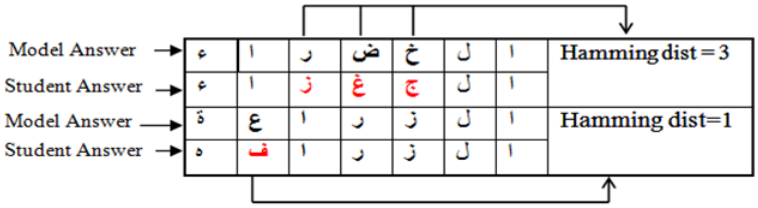
5.1 Experiment one

Evaluating candidate Answers based on derivative lexical similarity and space padding for the Arabic language

- **Hamming distance** is a metric for measure two binary data strings. While match two binary strings of the same length, Hamming distance is the number of bit positions in which the two bits are not the same. In the present work, hamming distance is applied to the data, which is not binary i.e. we have performed distance calculation based on the alphabet, present in the given word if both positions have a similar character the distance is zero else it is predicted to be one.

  As this word as an example:

  disthamming (الزارفة,الزراعة) = 1



Fig(2). Hamming Distance

- **The challenge in the hamming distance** is it only measures the distance between same length strings. If the lengths of the two strings or comparing strings are different the distance calculation is worthless. So, the standard Hamming model considers that the answer is only correct when both the answers (student answer and model answer) have the same number of characters and there is no any difference (missing, wrong and added characters) in any position. Standard Hamming model also not considers the answer is correct if both the student answer and model answer have lexical similarity and they are derivatives of the same root such as يلعب (play), لعب (played), لاعب (player for male), لاعبة (player for female), ملعب (Playground), لاعبون (players), لعب (playing). Also, there is another issue related to the article of the word ال (the) in the Arabic language, the article ال (the) is always attached with the Arabic word such as اللعب (the playing), the standard Hamming model considers the answer is wrong for the student answer اللعب (the playing)

that is not equal with model answer لعب (playing). The proposed Hamming model tried to solve these issues by applying the space padding pre-process to the answer that is smaller than the other answer. Padding space makes the text length of both answers is equal. Also, the proposed Hamming model tried to solve the similarity of unequal answer lengths by applying a space padding pre-process Also, the proposed Edit based model tried to solve issue of the lexical similarity for derivatives of the answer by applying another pre-process called stemming pre-process for both the student and model answers.

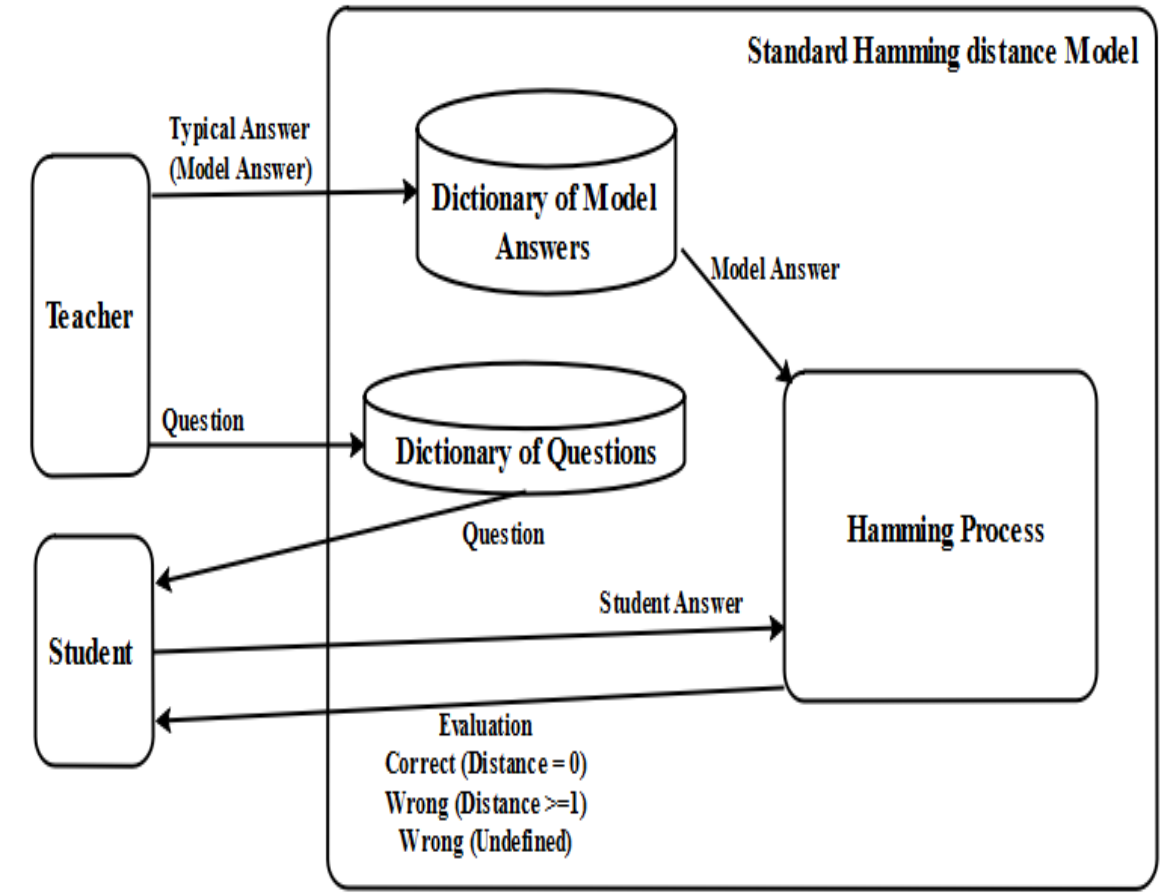


Fig (3). Standard Hamming Distance

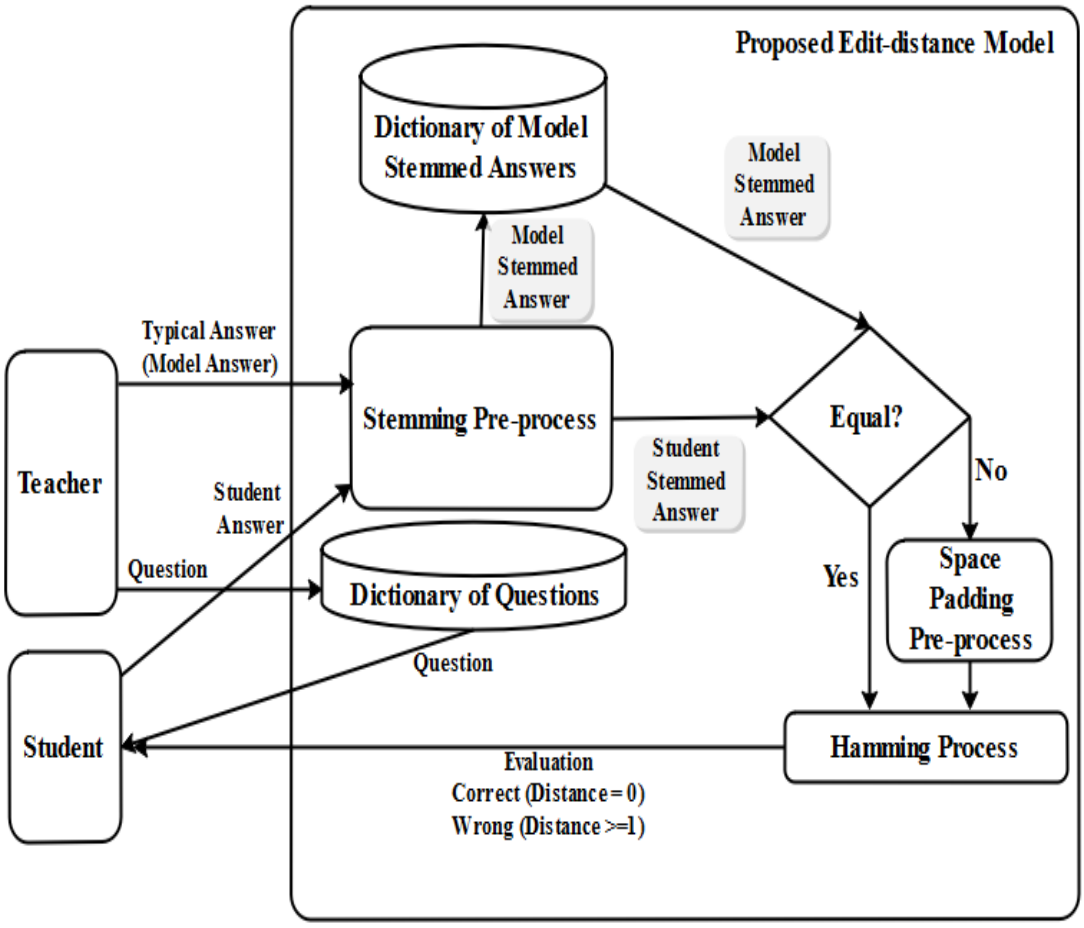


Fig (4). Proposed Edit Distance.

5.1.1 Data collection

To collect the required data for the proposed Hamming model, we designed some questions in Arabic language and stored in a dictionary of questions such as ماهي الحرفة التي كان يعمل بها اجدادنا القدماء في اليمن ؟ (What is the craft that our ancient ancestors used to work in Yemen?). We supposed the typical answer (model answer) for this question is الزراعة (farming), and we had many students answer the question (approximately 60 students), 20 students answered الزراعة (farming), 10 students answered مزرعة (farm), 7 students answered مزارعون (farmers), 3 students answered مزارع (farmer) , 10 students answered رياضة (sport) and 10 students answered زرا (the word is missing letters), and so on for other questions.

5.1.2 Pre-processing stage

- The proposed model used two pre-processes: stemming pre-process and space padding pre-process.
    - **The stemming pre-process** takes both the answers for the student and the model to remove the article ال (the) from the answer, and to return all derivatives of the word (the answer) to the root of the word. The derivatives of the answer have lexical similarity with the same meaning. The stemmed model answer will be stored in the dictionary of the model answers. The proposed model will use the root of the word to make the exactly lexical similarity between the student answer and typical answer.
    - **The space padding** is applied to the small length of the answer to becoming equal in length to the other answer. To understand the padding approach, we will investigate cases of standard Hamming distance model as follows:

There are three cases for the standard Hamming distance:

- The first case (normal case): when the lengths of both texts are equal. For example, text1 has 7 characters, and text2 has also 7 characters. So, the Hamming distance will have 2 parameters as follows:

Hamming (text1, text2)

The hamming operation will take the first parameter (text1) to compute its length (Here 7), and makes 7 comparisons between the characters of the two texts. The comparisons here are possible and sufficient to get the correct defined Hamming distance value between 0 to 7 because the text1 string is equal in length to the text2 string. So, there is no need for a space padding where both texts have the same length.

- The second case (issue case): when the lengths of both texts are un-equal. For example, text1 has 5 characters, and text2 has 7 characters. So, the Hamming operation will make 5 comparisons between the characters of the two texts. The comparisons here are possible but insufficient to get correct defined Hamming distance, where wrong defined Hamming distance value between 0 to 5 is produced, because the available comparisons (5 comparisons) are lesser than the text2 length (7 characters). So, there is a need for two space paddings for text1 to be both the texts have the same length (7 characters).

- The third case (second issue): Also, when the lengths of both texts are un-equal. For example, text1 has 7 characters, and text2 has 5 characters. So, the Hamming operation will make 7 comparisons between the characters of the two texts. The comparisons here are impossible, so undefined Hamming distance is occurred due to a string index out of range, where the required comparisons (7 comparisons) are greater than the text2 length (5 characters). So, there is a need for two space paddings for text2 to be both the texts have the same length (7 characters).

The space padding pre-process is not needed if both the answers have exactly the same root characters, but it is necessary if both the answers have different root characters.

5.1.3 Processing Stage

The main process of the proposed process is the Hamming process that takes both the answers of the student and the model as stemmed answers without space padding in case of the same root resulted from derivatives of the typical answer, and with space padding in case of different roots resulted from different answers to perform the Hamming operation for strings. Therefore, there is no possibility that the two answers are different in length; also, in most cases, the proposed model returns 0 value as a distance for derivative words. The proposed model gives equal or greater than 1 as a distance if there is missing or/and wrong in characters.

5.1.4 Algorithm of the proposed Hamming model:

Step 1: Give the question to the student from the questions dictionary.

Step 2: Take the student and model answers and perform the Stemming process on them.

Set st = ISRIStemmer()

Perform stdent_answer = st.stem(input("Answer: "))

Perform Stemming operation manually for model answer

and stored in model answer dictionary.

Step 3: perform the space padding on the stemmed answers

```
if len(stdent_answer)>len(value):
    m=len(stdent_answer)-len(value)
    value=value+(" "*m)
if len(stdent_answer)<len(value):
    m=len(value)-len(stdent_answer)
    stdent_answer=stdent_answer+(" "*m)
```

Step 4: Perform Hamming operation on the stemmed padded answers

```
r=hammingDist(stdent_answer,value)
```

Step 5: Hamming distance returns 0 value for correct answer and >=1 for the wrong answer

---

6. **Experiment two**

Development of an Arabic HQAS-Based ASAG to Consider an Ignored Knowledge in Misspelled Multiple Words Short Answers.

6.1 Proposed mothed

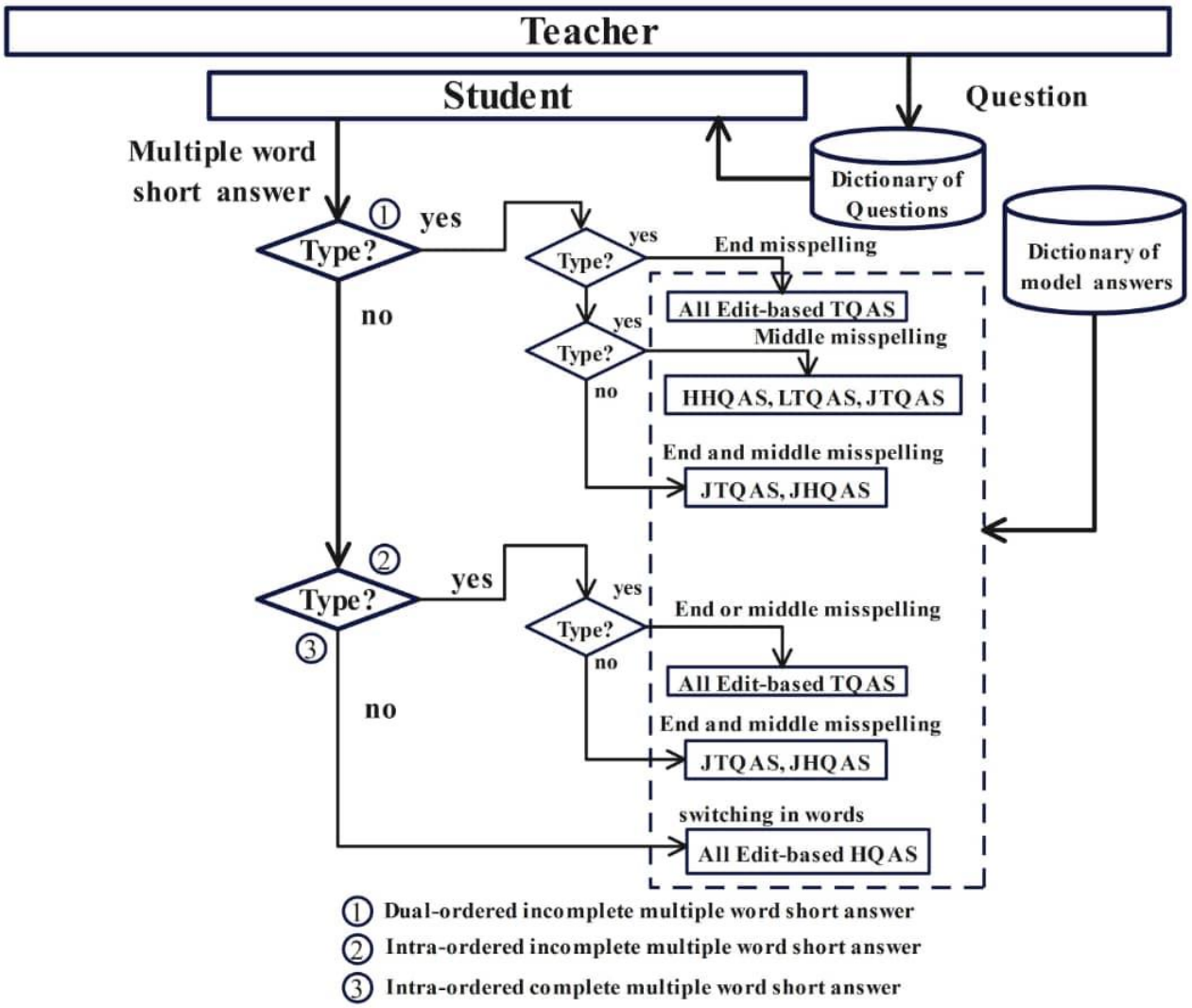


Fig (5). Hierarchical Question Answering System (HQAS).

One of the most common problems that are faced by many traditional questions answering systems (TQAS) is how to determine the precise cognitive performance for any student, where the questions of TQAS are organized in a flat structure that means all the questions are represented in one level where all the questions regardless of its complexity are conducted randomly by the students. Most multiple word-based short answers with TQAS are characterized by Fuzzy logic which means the answers may be not correct completely or not wrong completely, where the answers are close to the ideal correct or ideal wrong. Most TQAS programs based on ASAG(Automatic Short Answer Grading) are Boolean-based programs that consider the assessment wrong for fuzzy logic-based short answers. Boolean-based TQAS only considers the assessment correct if the edit similarity is 100%, and considers the assessment wrong if the edit similarity is not 100%. Although the student may have significant knowledge about the answer, but the teachers with TQAS can't differentiate between the student with high knowledge and the student with low knowledge if the assessments of the fuzzy logic answers for both the students are wrong. Also, TQAS can't keep track of the knowledge performance of the students, where teachers can't know which areas in the topic that most of the students couldn't understand clearly. To solve these issues, we have introduced a proposed model called Hierarchical Question Answering System (HQAS) in which questions are conducted based on a hierarchical traversing mechanism using a Breadth-First Search (BFS) algorithm on different levels instead of the flat traversing on one level. The hierarchical traversing mechanism is based on an m-ary tree. BFS starts the traversing from the root of the m-ary tree (for example node1) at level 0 and continues to traverse through children's nodes at level 1 (node 2 and node 3), then traverse through grandchildren nodes at level 2 (nodes 4 to node 8) as shown in fig 1. To differentiate between the students that have low knowledge and the students that have high knowledge on their fuzzy based answers that are considered the assessment wrong by any Boolean-based TQAS, we have implemented a knowledge assessment by considering their fuzzy-based answers into one of the assessments: very good, good, moderate, low, or very low. To determine which areas of the topic the students couldn't understand clearly, we have implemented the ideas of the topic as depths, each depth represented as a node connected only with their children such as depth 1 that includes node 1, node 2, and node 3 as shown in fig.1. By evaluating the answers of the student in each depth, we have gotten the knowledge performance in each area of the topic.

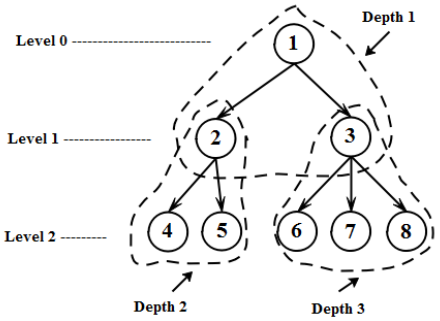


Fig (6). m-ary tree with eight nodes with three depths

In this paper, we have studied multiple words short answers (for example 2 words to 3 words) such as وحدة تحكم (control unit) or وحدة معالجة مركزية (central processing unit). The multiple word short answer may be a complete answer (answer with complete characters) or incomplete (answer with incomplete characters). Also, the multiple words short answers may be dual-ordered (intra-ordered and inter-ordered) such as وحدة تحكم (control unit) where the characters are ordered inside each word and the order between the two words are in order, or it may be only intra-ordered such as تحكم وحدة (unit control) where the characters inside both the words are in order, but the order between the two words is not in order, or it may be only inter-ordered such as ودحة تكحم (cotnrol uint) where the characters inside both the words are not in order, but the order between the two words is in order. Also, the multiple word short answer may be unordered such as تكحم ودحة (uint control) where the characters inside both the words are not in order, and also the order between the two words is not in order. All the types of complete and incomplete multiple word short answers can be classified as shown in fig 2.

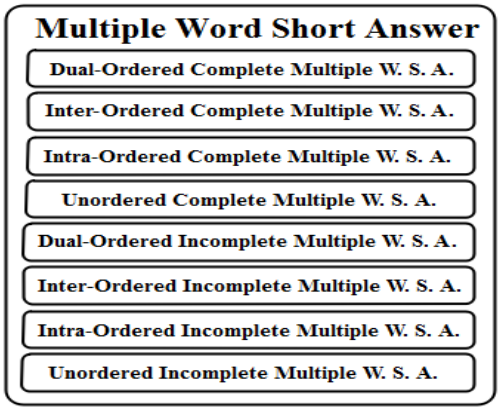
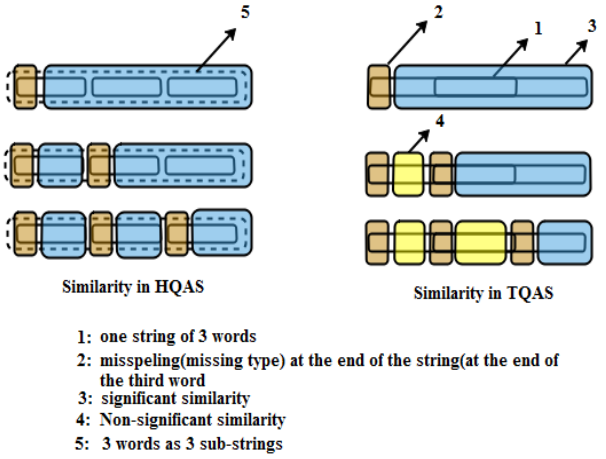

Multiple Word Short Answer
Dual-Ordered Complete Multiple W. S. A.
Inter-Ordered Complete Multiple W. S. A.
Intra-Ordered Complete Multiple W. S. A.
Unordered Complete Multiple W. S. A.
Dual-Ordered Incomplete Multiple W. S. A.
Inter-Ordered Incomplete Multiple W. S. A.
Intra-Ordered Incomplete Multiple W. S. A.
Unordered Incomplete Multiple W. S. A.

Fig (7). Types of complete and incomplete multiple words short answers

From all these types, we have used only types of dual and Intra ordered multiple word short answers, where the meaning of the answer concept is still kept. Our paper has addressed the ignored knowledge (hidden knowledge) of the student resulting from the TQAS assessment wrong for all fuzzy-based answers.

The TQAS assessment wrong is given to any misspelled answer of either missing or substituting type. We have done a comparison between the TQAS and the proposed HQAS using three common edit-based algorithms: Hamming, Levenshtein, and Jaro_Winkler. A TQAS mechanism of any edit-based algorithm considers only one string for all words of the answer which means a low similarity can be occur for small misspelling in the middle of the string, for example, in Hamming-based TQAS, small misspelling of missing type at the end of any word in the middle of the string for the dual-ordered incomplete multiple word short answer can cause a high non-significant similarity after the misspelling and low significant similarity before the misspelling. We named non-significant similarity because there is a similarity but not considered by the algorithm due to the different locations of the similar characters where the Hamming algorithm has only substituting-ability and can't address the missing without adding ability. To solve the problem in Hamming-based TQAS, there are many ways either we use other types of edit-based algorithms such as Levenshtein and Jaro_Winkler or we use a modified Hamming by implementing a proposed Hamming-based HQAS that considers each word as one string, so the answer will contain many strings, this mechanism will decrease the amount of the non-significant similarity and increase the significant similarity as shown in the figure 3. To still keep the content of the word, our paper has been focused on the missing type that occurs at the end of the string to evaluate an accepted level of the knowledge about the answer. To understand the mechanism of the algorithm in HQAS, suppose the model answer composed of words a1, a2..., an, and the student answer composed of words b1, b2..., bm, then the similarity can be calculated by the average of the sum of the maximum similarities between the words of the model answer and the words of the student answer as shown in the following proposed equation.

$$\text{HQAS\_based Simalarity} = \frac{\sum_{i=1}^{n} \max[\text{sim}(a_i, b_1), \text{sim}(a_i, b_2), \dots, \text{sim}(a_i, b_m)]}{n} \qquad (1)$$



Fig(8). Hamming-based similarity between TQAS and HQAS in dual-ordered incomplete multiple word short answer.

Also, the same problem can occur with the type of the answer: intra-ordered complete multiple word short answer, where all the words of the answer are correct and complete but some or all the words are not in the right place, causing a misspelling of word-based substituting type such as the student answer تحكم وحدة (unit control) related to model answer وحدة تحكم (control unit), so this type of the answer produces very high non-significant similarity because the misspelling based on the changing in location for all the characters of any word in the answer. This type of answer carries a big amount of knowledge but the three common edit-based TQAS algorithms: Hamming, Levenshtein, and Jaro_Winkler don't take into account this knowledge. To solve this problem with this type of the answer, we have been used the edit-based HQAS including these algorithms that their mechanisms are based on the proposed equation 1 to increase the amount of the significant similarity and remove any non-significant similarity from the answer as shown in fig 8.

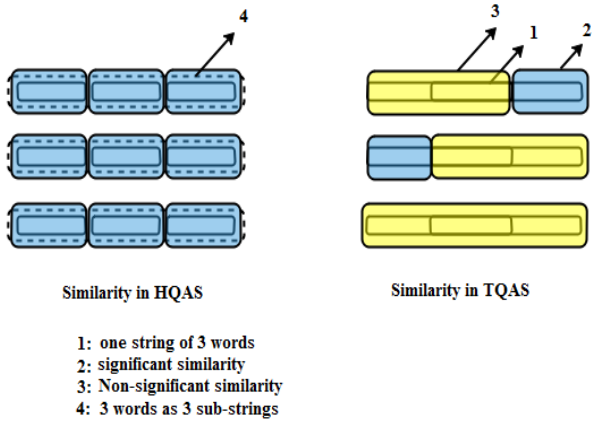
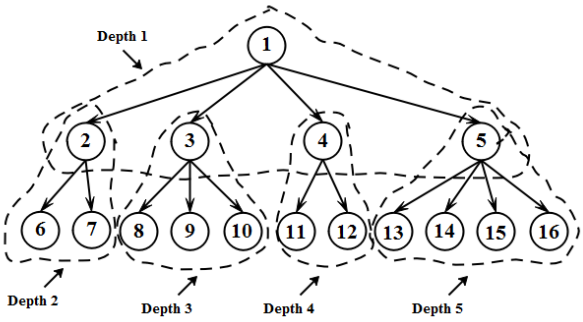


Fig (9). Character similarity based on Hamming, Levenshtein, and Jaro_Winkler algorithms between TQAS and HQAS in intra-ordered complete multiple word short answer.

To investigate the difference in performance between the proposed HQAS model and the TQAS model using Hamming, Levenshtein, and Jaro_Winkler. We have been implemented questions of a certain exam about any topic (for example 16 questions about components of a computer), and we have been collected the data of the student's answer of the types: dual-ordered and intra-ordered incomplete and complete multiple words short answers that carry the knowledge close to the model answer. We have done two scenarios. In the first scenario, TQAS is implemented using the mentioned edit-based algorithms, where all the 16 questions are conducted randomly in one level (flat level) by the students. In the second scenario, HQSA is implemented as an m-ary tree, where the 16 questions are conducted by the students sequentially in multiple levels (heretical levels). In this example, three levels are used, and five depths are drawn in the tree to represent five ideas of the topic as shown in figure 9.



Fig(10). m-ary tree with 16 nodes with 5 depths.

As shown in tables 3,4 and 5, and figures 13, 14, and 15 all related statistical percentages are obtained for the similarity and knowledge assessment for the answer of a certain question by conducting TQAS and proposed HQAS using Hamming algorithm between the model answer and the student answer, where Hamming-based HQAS model has been achieved high percentage similarity and high knowledge assessments than Hamming-based TQAS model when the misspelling occurred at the middle of the dual-ordered incomplete answer, intra-ordered incomplete answer and intra-ordered complete answer. but there is a similar percentage of the similarity and knowledge assessments in both models when the misspelling occurred at the end of the dual-ordered incomplete answer or when there is no misspelling in the dual-ordered complete answer. Also, Levenshtein-based HQAS and Jaro_Winkler-based HQAS have been achieved high percentage similarity and high knowledge assessments than Levenshtein-based TQAS and Jaro_Winkler-based TQAS model respectively when the misspelling occurred in the intra-ordered incomplete answer and intra-ordered complete answer, but there is a similar percentage of similarity and knowledge assessment in both models when the misspelling occurred in the dual-ordered incomplete answer and dual-ordered complete answer. To investigate the performance of the proposed edit-based HQAS on the 16 questions to determine the knowledge performance of the students on different areas of the topic, we have noticed in

figure 9, the knowledge performance of the students is increased in the depth 1, depth 2 and depth 3, but it is decreased in depth 5, and it was moderate at depth 4.

7. Experiment Three

Deep learning based on question answering system in Arabic language.

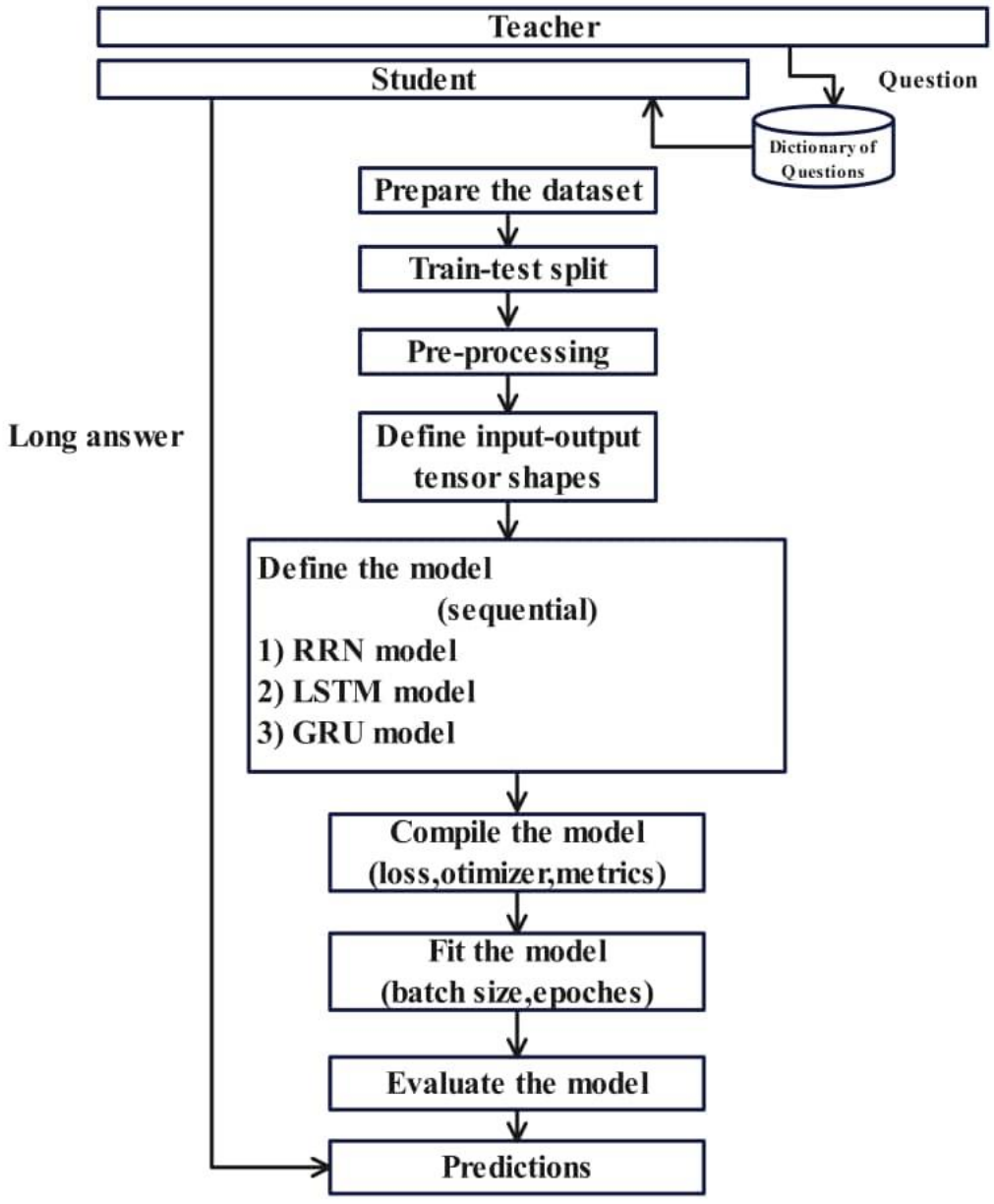


Fig (11).deep learning model

There are millions pieces of data available, but making the right information accessible when needed is very important. Getting the right documents to read and further getting a direct answer to one's question from the set of documents is a challenging task. A question answering system is concerned with building systems that automatically answer questions posed by humans in a natural language. A QA system is generally programmed to pull answers from a structured database or an unstructured collection of natural language document.

- **A recurrent neural network (RNN)** is an artificial neural network where connections between units form a directed cycle 1. An RNN can deal with variable-length sequence input. An RNN is a single model used to summarizing information from the past and providing it to the next stage. Inputs at each stage Xi and the summary of all events till now coming from the previous stage. Outputs are the predictions and connects to the next stage (i.e., provides a summarized information of the past to the next stage.). RNN suffers from a few drawbacks like not exploiting the future context and memory loss due to the Vanishing Gradient Problem.
- **Long Short-Term Memory (LSTM)** is RNN that are used to deal with variable length continuous sequence inputs. These are specialized to carry history in long sequence of input. LSTMs do this by storing the recurrent hidden vector which is dependent previous hidden vector. LSTM is one of the popular RNN techniques to mitigate the vanishing gradient problem of RNN [23].
- **Gated Recurrent Unit (GRU)** is a type of Recurrent Neural Network that addresses the issue of long term dependencies which can lead to vanishing gradients larger RNN networks experience. GRUs addresses this issue by storing "memory" from the previous time point to help inform the network for future predictions. Gated Recurrent Unit is a type of Recurrent Neural Network that addresses the issue of long term dependencies which can lead to vanishing gradients larger RNN networks experience. GRUs addresses this issue by storing "memory" from the previous time point to help inform the network for future predictions.

7.1 Prepare Dataset

We have used questions are answered by students from grade 4 to grade 9. We have acquired this dataset from the Arabic subject books. This dataset consists 4 sets of type's answers which are written by students. The dataset consists of around 11000 answers. The dataset we get is a cvs file, a tab separated value file consisting of answers label, answers set.

7.2 Collecting Data

We have acquired our data from different Arabic books (as mentioned in the above section: Dataset). We have collected a total of around 11000 on subjective answering type (single word short answer, multiple word short answer, long answer).

7.3 Data preprocessing

- We used NLTK (Natural Language Toolkit), a Python computer language package, and more additional libraries to support Arabic text in preprocess the data and neural network.
- We first eliminate all integers, whitespaces, and default stop words during data preprocessing. Stop words are words that provide no functional use in a sentence. Therefore, it seems sense to get rid of them as they don't add much to the meaning. The sentence was then divided into tokens after cleaning. We then take features from this, such as word count, character count, average word length, misspelt words, and POS tagging.
- To produce feature vectors, we have used model architecture from word2vec called (Skip-Gram) which takes in text corpus as input and pops out feature vectors as its output.. Word2vec model creates vectors which might be allotted numerical representations of word functions, features consisting of the context of character words.

7.4 Building the Models

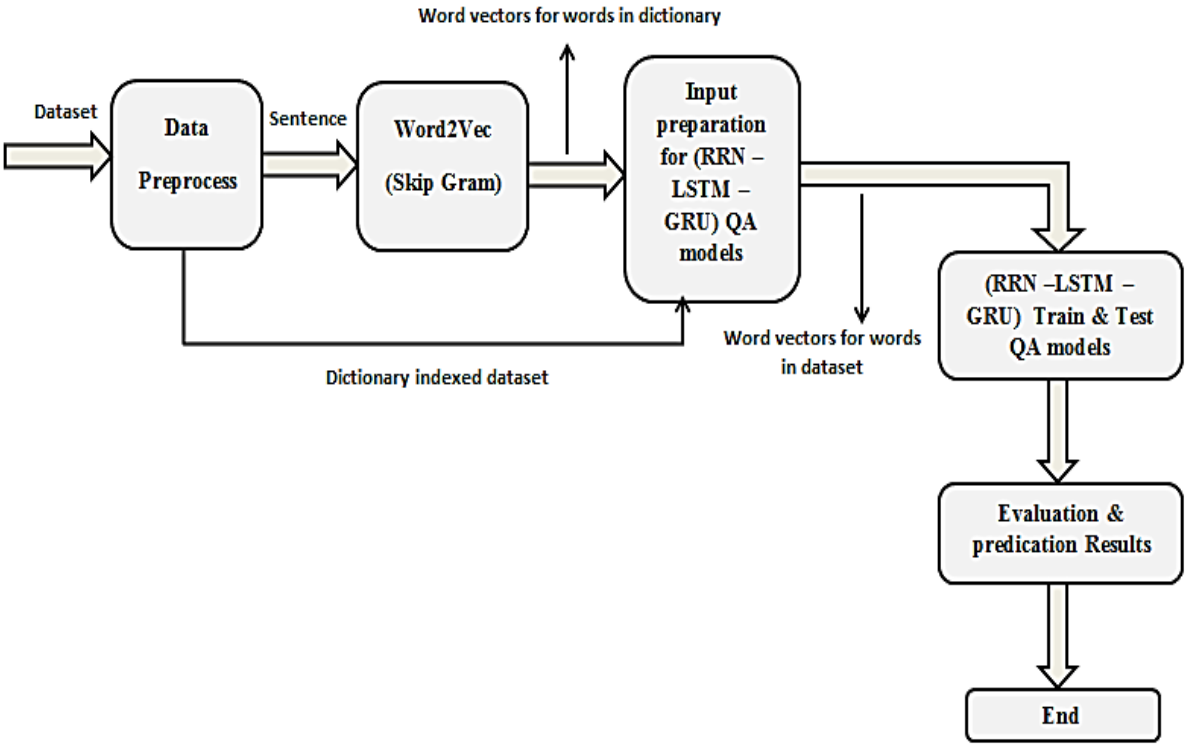

Word vectors for words in dictionary
Dataset
Data Preprocess
Sentence
Word2Vec (Skip Gram)
Input preparation for (RRN – LSTM – GRU) QA models
(RRN –LSTM – GRU) Train & Test QA models
Word vectors for words in dataset
Dictionary indexed dataset
Evaluation & predication Results
End

Fig (12). Flowchart AQAS deep learning models

There are important steps for building a RNN – LSTM – GRU deep neural network

1. This model can be built as a tf.keras.Sequential.
2. The first layer is the encoder, which converts the text to a sequence of token indices.

3. After the encoder is an embedding layer. An embedding layer stores one vector per word. When called, it converts the sequences of word indices to sequences of vectors. These vectors are trainable. After training (on enough data), words with similar meanings often have similar vectors.This index-lookup is much more efficient than the equivalent operation of passing a one-hot encoded vector through a tf.keras.layers.Dense layer.
4. recurrent neural network (RNN) processes sequence input by iterating through the elements. RNNs pass the outputs from one timestep to their input on the next timestep.

5. After the RNN has converted the sequence to a single vector the two 'layers.Dense' do some final processing, and convert from this vector representation to a single logit as the classification output.

6. The LSTM model helps alleviate the Vanishing Gradient problem of the Recurrent NeuralNetworks hence help provide a better memory. This is achieved with the help of gates.Given an input sequence x = {x(1), x(2),.....,x(n)}, where x(t) is an E-dimension word vector. The equations for the gates in LSTM are:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$

$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

$i_t \rightarrow represents\ input\ gate.$
$f_t \rightarrow represents\ forget\ gate.$
$o_t \rightarrow represents\ output\ gate.$
$\sigma \rightarrow represents\ sigmoid\ function.$
$w_x \rightarrow weight\ for\ the\ respective\ gate(x)\ neurons.$
$h_{t-1} \rightarrow output\ of\ the\ previous\ lstm\ block(at\ timestamp\ t-1).$
$x_t \rightarrow input\ at\ current\ timestamp.$
$b_x \rightarrow biases\ for\ the\ respective\ gates(x).$

6. A gated recurrent unit (GRU) used less memory and is faster than LSTM. However, LSTM is more accurate when using datasets with longer sequences.

8. Results

8.1 Experiment one

The result of the proposed edit based model is shown in the table below. The standard Hamming model achieved 33.3% correct answer with 0 correct defined distance value, and achieved 58.3% wrong answer with greater than or equa1 to 1 wrong defined distance value, and achieved 8.3% wrong answer with undefined distance value. But, The proposed edit based model achieved 75%

correct answer with 0 correct defined distance value, and achieved 25% wrong answer with greater than or equa1 to 1 correct defined distance value.

Table (2). Comparison between standard Hamming method and proposed Hamming method.

ماهي الحرفة التي كان يعمل بها اجدادنا القدماء في اليمن ؟ (question) model answer: الزراعة (7 characters) (farm)					
Number of Students	Student answer	Distance		Evaluation	
		standard Hamming model	proposed Edit based model	standard Hamming model	proposed Edit based model
20	الزراعة (7 characters)	0	0	Correct	Correct
10	مزرعة (5 characters)	5	0	Wrong	Correct
7	مزارعون (7 characters)	6	0	Wrong	Correct
3	مزارع (5 characters)	4	0	Wrong	Correct
5	رياضة (5 characters)	5	3	Wrong	Wrong
10	زرا (3 characters)	3	1	Wrong	Wrong
5	المزروعات (8 characters)	Undefined	0	Wrong	Correct

Fig (13). distance of the standard Hamming model

Fig (14). distance of the proposed edit based model

8.2 Experiment Two

As shown in tables 3,4 and 5, and figures 14, 15, and 16, all related statistical percentages are obtained for the similarity and knowledge assessment for the answer of a certain question by conducting TQAS and proposed HQAS using Hamming algorithm between the model answer and the student answer, where Hamming-based HQAS model has been achieved high percentage similarity and high knowledge assessments than Hamming-based TQAS model when the misspelling occurred at the middle of the dual-ordered incomplete answer, intra-ordered incomplete answer and intra-ordered complete answer. but there is a similar percentage of the similarity and knowledge assessments in both models when the misspelling occurred at the end of the dual-ordered incomplete answer or when there is no misspelling in the dual-ordered complete answer. Also, Levenshtein-based HQAS and Jaro_Winkler-based HQAS have been achieved high percentage similarity and high knowledge assessments than Levenshtein-based TQAS and Jaro_Winkler-based TQAS model respectively when the misspelling occurred in the intra-ordered incomplete answer and intra-ordered complete answer, but there is a similar percentage of similarity and knowledge assessment in both models when the misspelling occurred in the dual-ordered incomplete answer and dual-ordered complete answer. To investigate the performance of the proposed edit-based HQAS on the 16 questions to determine the knowledge performance of the students on different areas of the topic, we have noticed in figure 9, the knowledge performance of the students is increased in the depth 1, depth 2 and depth 3, but it is decreased in depth 5, and it was moderate at depth 4.

Table 3: character-based similarity and acknowledge assessment in both TQAS and HQAS using Hamming algorithm

	Question:ماهي الاجزاء التي يتكون منها الحاسب وتستطيع لمسها ومشاهدتها؟						
	Model Answer: مكونات مادية						
Student answer	Type of answer	Location of missing in the answer	No. of Students	Hamming-based TQAS		Hamming-based HQAS	
				Similarity	knowledge assessment	Similarity	knowledge assessment
مكونات مادي	dual-ordered incomplete answer	at the end	12	91.60%	very good	91.60%	very good
مكونات ماد			11	83.30%	good	83.30%	good
مكونات ما			9	75.00%	moderate	70.00%	moderate
مكونا مادية	dual-ordered incomplete answer	at the middle	15	41.60%	low	91.60%	very good
مكون مادية			11	33.30%	very low	83.30%	good
مكو مادية			8	25.00%	very low	75.00%	moderate
مكونا مادي	dual-ordered incomplete answer	at the middle and the end	12	41.60%	very low	91.60%	very good
مكون ماد			10	33.30%	very low	63.60%	low
مكو ما			6	25.00%	very low	45.00%	very low
مادية مكونا	intra-ordered incomplete answer	at the end	11	8.30%	very low	91.60%	very good
مادية مكون			9	8.30%	very low	83.30%	good
مادية مكو			8	8.30%	very low	75.30%	moderate
مادي مكونات	intra-ordered incomplete answer	at the middle	12	8.30%	very low	90.00%	very good
ماد مكونات			9	8.30%	very low	80.00%	good
ما مكونات			4	8.30%	very low	70.00%	moderate
مادي مكونا	intra-ordered incomplete answer	at the middle and the end	7	8.30%	very low	81.60%	good
ماد مكون			5	8.30%	very low	63.30%	low
ما مكون			4	8.30%	very low	45.00%	very low
مكونات مادية	dual-ordered complete answer	null	15	100.00%	very good	100.00%	very good
مادية مكونات	intra-ordered complete answer	switching in words	9	8.30%	very low	100.00%	very good

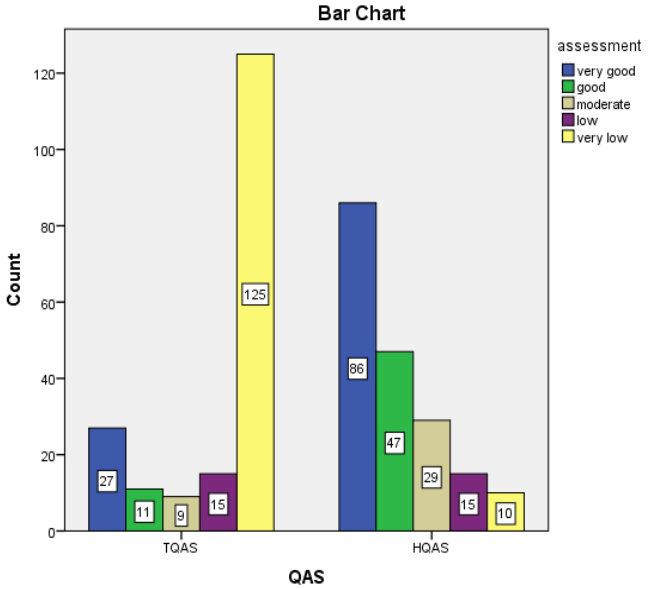


Fig (15).Difference in the hidden acknowledge assessments on both TQAS and HQAS using Hamming algorithm

Table 4: character-based similarity and acknowledge assessment in both TQAS and HQAS using Levenshtein algorithm

	Question: ماهي الاجزاء التي يتكون منها الحاسب وتستطيع لمسها ومشاهدتها؟						
	Model Answer: مكونات مادية						
Student answer	**Type of answer**	**Location of missing in the answer**	**No. of Students**	**Levenshtein -based TQAS**		**Levenshtein -based HQAS**	
				Similarity	**knowledge assessment**	**Similarity**	**knowledge assessment**
مكونات مادي	dual-ordered incomplete answer	at the end	12	91.66%	very good	90.00%	very good
مكونات ماد			11	83.33%	good	80.00%	good
مكونات ما			9	75.00%	moderate	70.00%	moderate
مكونا مادية	dual-ordered incomplete answer	at the middle	15	91.66%	very good	91.66%	very good
مكون مادية			11	83.33%	good	83.33%	good
مكو مادية			8	75.00%	moderate	75.00%	moderate
مكونا مادي	dual-ordered incomplete answer	at the middle and the end	12	83.33%	good	81.66%	good
مكون ماد			10	66.66%	low	63.33%	low
مكو ما			6	50.00%	very low	45.00%	very low
مادية مكونا	intra-ordered incomplete answer	at the end	11	25.00%	very low	**91.66%**	very good
مادية مكون			9	25.00%	very low	**83.33%**	good
مادية مكو			8	25.00%	very low	**75.00%**	moderate
مادي مكونات	intra-ordered incomplete answer	at the middle	12	16.66%	very low	**90.00%**	very good
ماد مكونات			9	25.00%	very low	**80.00%**	good
ما مكونات			4	25.00%	very low	**70.00%**	moderate
مادي مكونا	intra-ordered incomplete answer	at the middle and the end	7	25.00%	very low	**81.66%**	good
ماد مكون			5	33.33%	very low	**63.33 %**	low
ما مكون			4	33.33%	very low	**53.33%**	very low
مكونات مادية	dual-ordered complete answer	null	15	100.00%	very good	100.00%	very good
مادية مكونات	intra-ordered complete answer	null	9	16.66 %	very low	**100.00 %**	very good

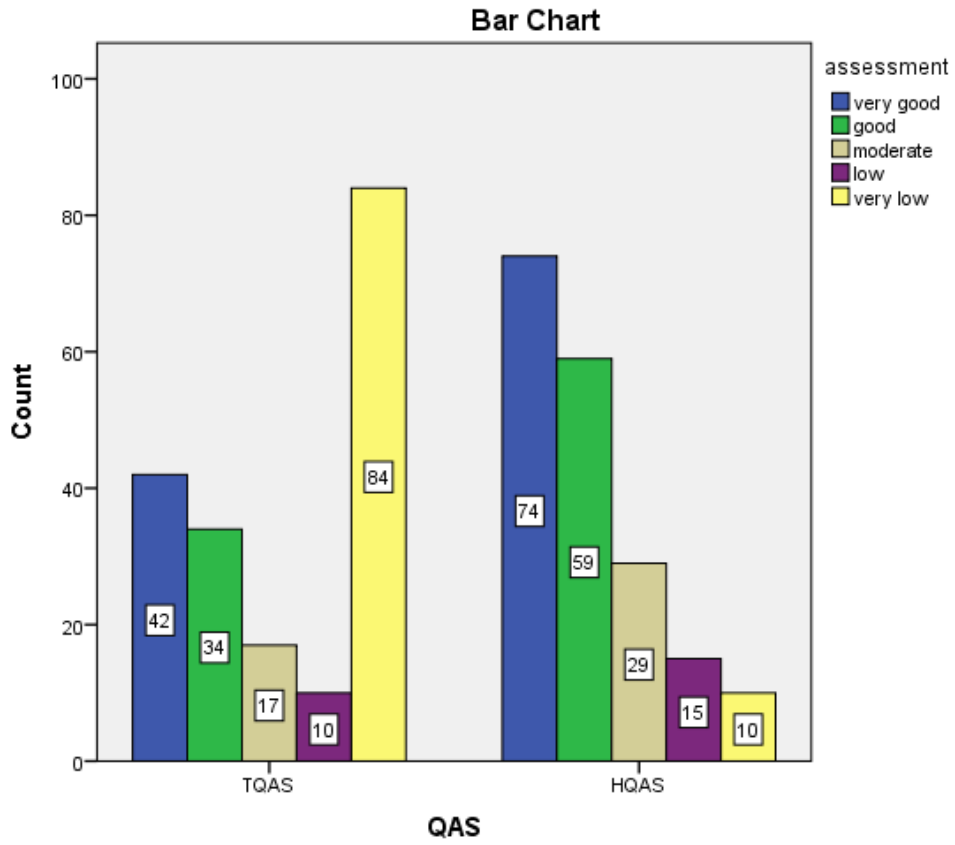


Fig (16). Differences in the hidden acknowledge assessments on both TQAS and HQAS using Levenshtein algorithm.

Table 5: Character-based similarity and acknowledge assessment in both TQAS and HQAS using Jaro_Winkler algorithm

	Question:؟ماهي الاجزاء التي يتكون منها الحاسب وتستطيع لمسها ومشاهدتها						
	Model Answer: مكونات مادية						
Student answer	Type of answer	Location of missing in the answer	No. of Students	Jaro_Winkler-based TQAS		Jaro_Winkler-based TQAS	
				Similarity	knowledge assessment	Similarity	knowledge assessment
مكونات مادي	dual-ordered incomplete answer	at the end	12	98.33%	very good	98.00%	very good
مكوناتماد			11	96.66%	very good	93.33%	very good
مكونات ما			9	95.00%	very good	90.00%	very good
مكونا مادية	dual-ordered incomplete answer	at the middle	15	98.33%	very good	98.33%	very good
مكون مادية			11	94.66%	very good	96.66%	very good
مكو مادية			8	91.57%	very good	91.66%	very good
مكونا مادي	dual-ordered incomplete answer	at the middle and the end	12	96.66%	very good	96.33%	very good
مكون ماد			10	90.83%	very good	90.00%	very good
مكو ما			6	84.44%	good	81.66%	good
مادية مكونا	intra-ordered incomplete answer	at the end	11	62.37%	low	98.33%	very good
مادية مكون			9	57.77%	very low	96.66%	very good
مادية مكو			8	59.25%	very low	91.66%	very good
مادي مكونات	intra-ordered incomplete answer	at the middle	12	73.71%	moderate	98.0%	very good
ماد مكونات			9	76.16 %	moderate	93.33%	very good
ما مكونات			4	79.16%	moderate	90.00%	very good
مادي مكونا	intra-ordered incomplete answer	at the middle and the end	7	69.72%	low	96.33%	very good
ماد مكون			5	67.65%	low	90.00%	very good
ما مكون			4	74.64 %	moderate	86.66%	good
مكونات مادية	dual-ordered complete answer	null	15	100.00 %	very good	100.00 %	very good
مادية مكونات	intra-ordered complete answer	null	9	61.11%	very low	100.00 %	very good

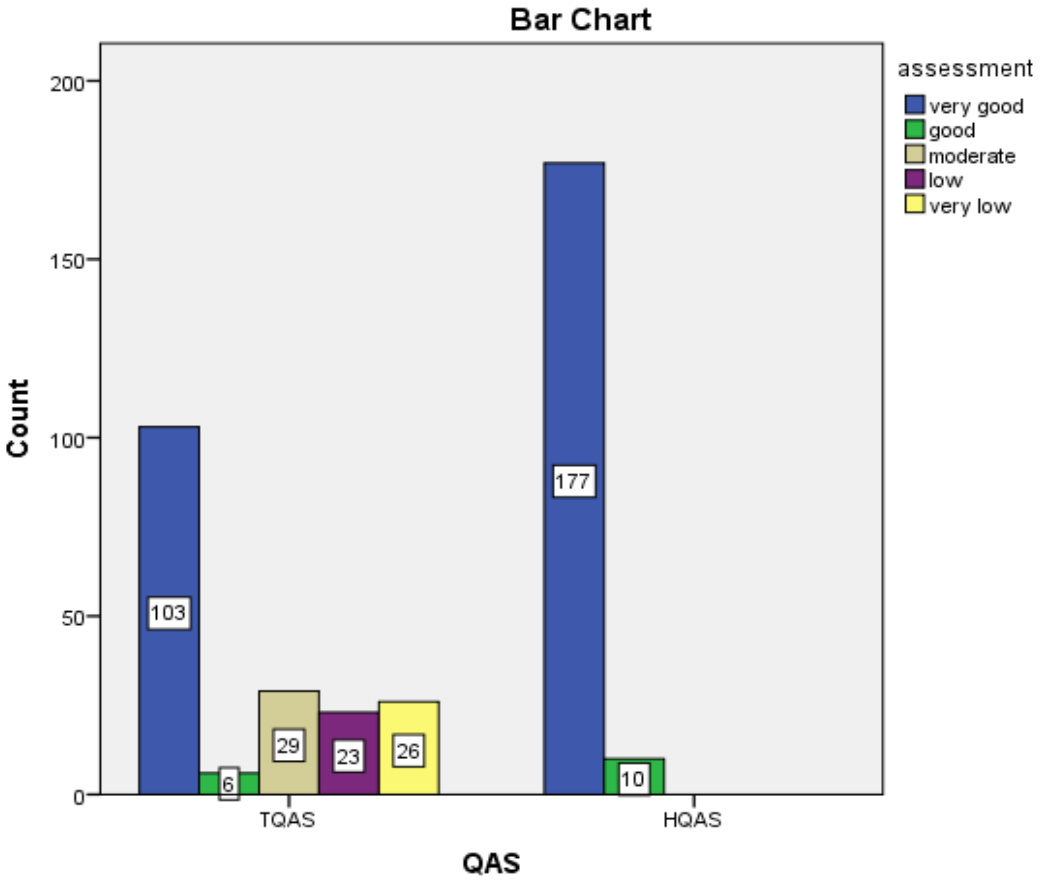
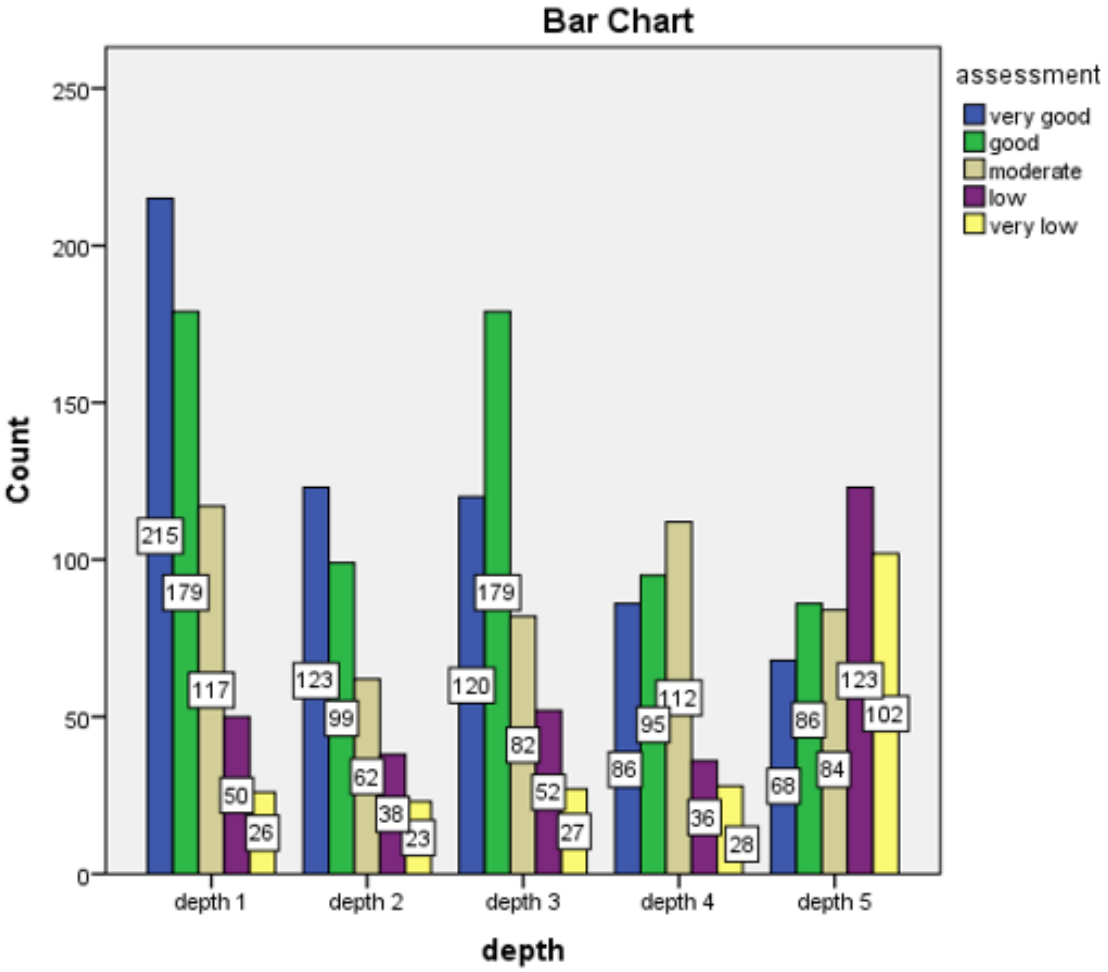


Fig (17). Differences in the hidden acknowledge assessments on both TQAS and HQAS using Jaro_Winkler algorithm.



Fig(18). Distribution of acknowledge performance on different depths of the topic.

8.2.1 Accuracy Measurements

To measure the accuracy, precision, recall, and F1 factor between the TQAS and the proposed HQAS using Hamming, Levenshtein, and Jaro Winkler, we have been implemented a truth of the student answers (actual answers) of different types of the answer using the root of the Arabic word to extract the important characters that indicate to the significant acknowledge, so we named akn (has knowledge) to indicate the positive truth, and we named no akn (has no aknwledge) to indicate the negative truth. The 20 student answers for a certain question are used as an example to extract the accuracy measurements between the two models as shown in table 4. The equations of the accuracy measurements are as follows:

$$\text{Accuracy} = \frac{T_{akn} + T_{non\_akn}}{T_{akn} + F_{non\_akn} + T_{non\_akn} + F_{akn}} \quad (2)$$

$$\text{Precision(akn)} = \frac{T_{akn}}{T_{akn} + F_{akn}} \quad (3)$$

$$\text{Recall(akn)} = \frac{T_{akn}}{T_{akn} + F_{non_{akn}}} \quad (4)$$

$$\text{F1(akn)} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

Table 6: The actual assessment for the acknowledge about the answer.

Answer	1	2	3	4	5	6	7	8	9	10
Truth	akn	akn	no_akn	akn	akn	akn	akn	akn	akn	akn
Answer	11	12	13	14	15	16	17	18	19	20
Truth	akn	akn	akn	akn	no_akn	akn	akn	no_akn	akn	akn

As shown in the table 7, HQAS has been achieved higher accuracy, precision, recall and F1-score than TQAS using Hamming, Levenshtein and Jaro_Winkler.

Table 7: Difference in accuracy, precision, recall and F1-score in both TQAS and HQAS using Hamming, Levenshtein and Jaro_Winkler algorithms

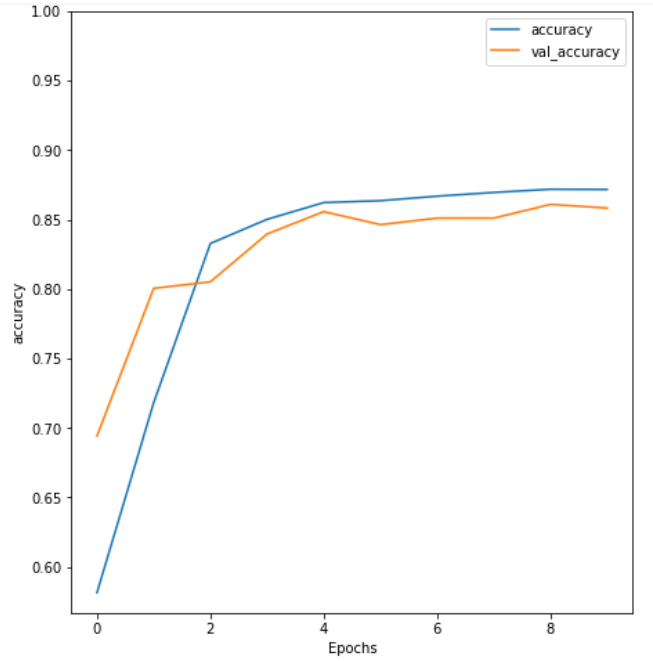
Algorithm	Model	classes	Precision	recall	F1-score	Accuracy
Hamming	TQAS	akn	0.75	0.19	0.3	0.3
		no_akn	0.19	0.75	0.3	
	HQAS	akn	0.88	0.88	0.88	0.7
		no_akn	0.5	0.5	0.5	
Levenshtein	TQAS	akn	0.88	0.44	0.59	0.5
		no_akn	0.25	0.75	0.38	
	HQAS	akn	0.88	0.88	0.88	0.8
		no_akn	0.5	0.5	0.5	
Jaro_Winkler	TQAS	akn	0.71	0.67	0.69	0.55
		no_akn	0.17	0.2	0.18	
	HQAS	akn	0.8	1	0.88	0.8
		no_akn	0	0	0	

8.3 Experiment Three

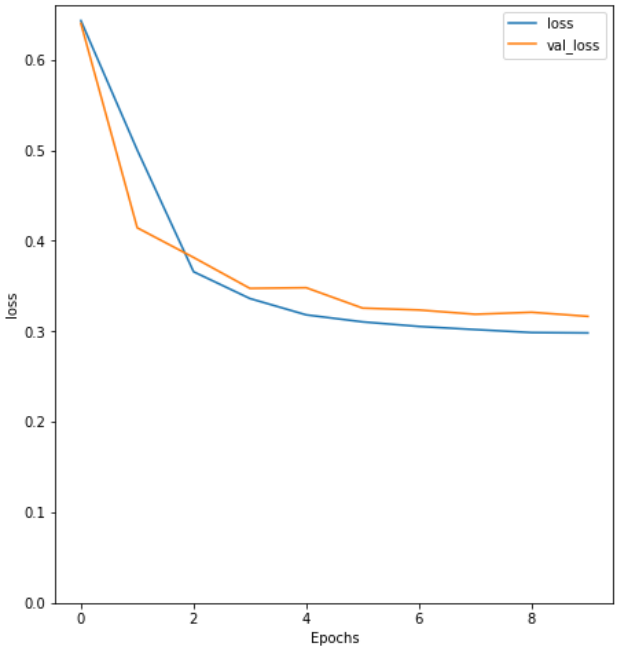
Table8: Performance metrics on RRN model .

RNN				
model.fit(input_train, y_train, epochs=10, validation_split=0.2)				
	loss	Accuracy	val_loss	val_accuracy
1	0.6432	0.5815	0.6400	0.6943
2	0.5002	0.7182	0.4143	0.8005
3	0.3658	0.8328	0.3816	0.8052
4	0.3362	0.8502	0.3474	0.8396
5	0.3181	0.8622	0.3481	0.8557
6	0.3104	0.8636	0.3256	0.8464
7	0.3054	0.8668	0.3235	0.8510
8	0.3019	0.8695	0.3188	0.8510
9	0.2986	0.8718	0.3210	0.8609
10	0.2982	0.8717	0.3165	0.8583
model.fit(input_train, y_train, epochs=20, validation_split=0.2)				
11	0.2983	0.8729	0.3256	0.8562
12	0.2971	0.8729	0.3275	0.8578
13	0.2948	0.8750	0.3224	0.8552
14	0.2955	0.8741	0.3219	0.8536
15	0.2928	0.8741	0.3219	0.8536
16	0.2930	0.8764	0.3225	0.8573
17	0.2902	0.8761	0.3222	0.8521
18	0.2883	0.8794	0.3200	0.8594
19	0.2890	0.8788	0.3308	0.8443
20	0.2867	0.8790	0.3192	0.8557
model.fit(input_train, y_train, epochs=50 , validation_split=0.2)				
21	0.2703	0.8858	0.3207	0.8594
22	0.2700	0.8873	0.3276	0.8573
23	0.2662	0.8894	0.3299	0.8583
24	0.2641	0.8898	0.3267	0.8672
25	0.2629	0.8896	0.3243	0.8625
26	0.2594	0.8926	0.3277	0.8510
27	0.2563	0.8940	0.3399	0.8411
28	0.2532	0.8973	0.3317	0.8557
29	0.2520	0.8947	0.3419	0.8573
30	0.2470	0.8989	0.3510	0.8620
31	0.2438	0.8998	0.3415	0.8490
32	0.2404	0.8940	0.3462	0.8521
33	0.2532	0.8973	0.3317	0.8557
34	0.2520	0.8989	0.3510	0.8620
36	0.2438	0.8998	0.3415	0.8490
37	0.2404	0.9003	0.3462	0.8521
39	0.2422	0.9014	0.3560	0.8536
38	0.2340	0.9058	0.3501	0.8516
40	0.2113	0.9170	0.3673	0.8432
41	0.2057	0.9188	0.3844	0.8562
42	0.2041	0.9201	0.3933	0.8458

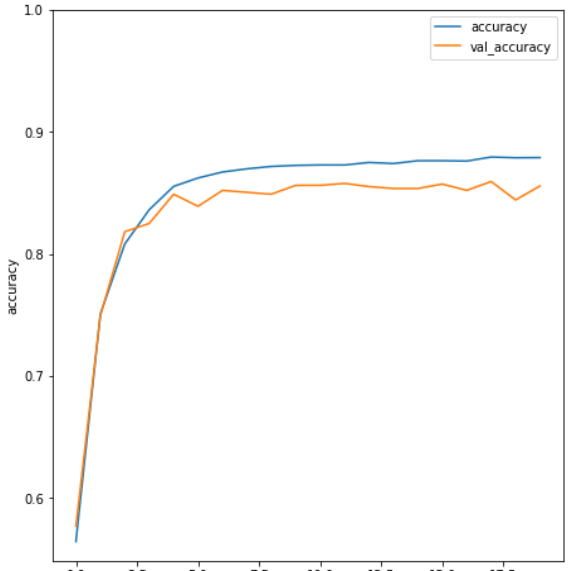
43	0.1988	0.9220	0.4165	0.8510
44	0.1942	0.9258	0.3970	0.8417
45	0.1885	0.9284	0.4029	0.8557
46	0.1897	0.9283	0.4141	0.8568
47	0.1832	0.9317	0.4061	0.8417
48	0.1821	0.9318	0.4274	0.8453
49	0.1712	0.9380	0.4307	0.8354
50	0.1680	0.9402	0.4373	0.8536



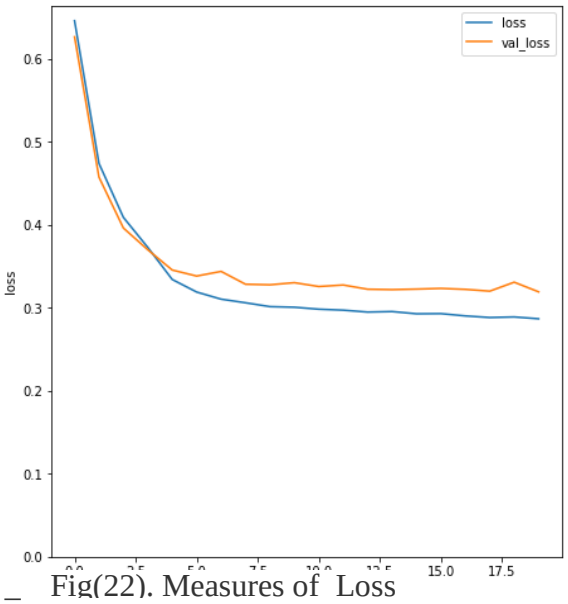
Fig(19). Measures of accuracy and val accuracy on the training set and the validation set on 10 epochs.



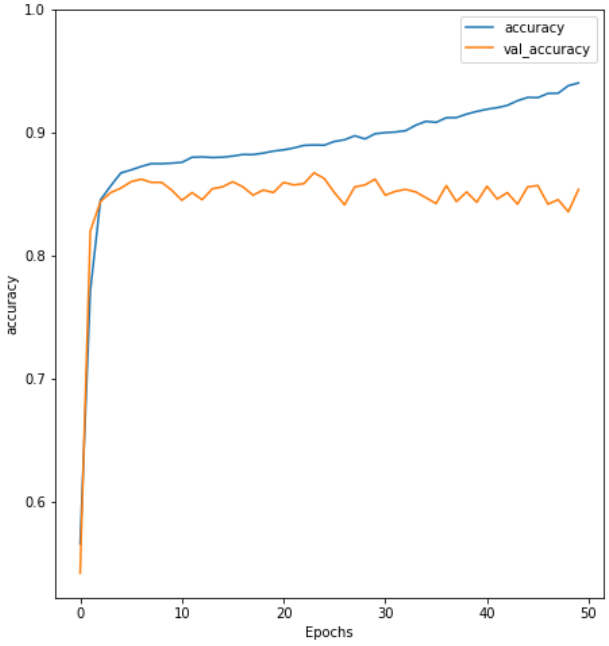
Fig(20). Measures of Loss and val loss on the training set and the validation set on 10epochs.



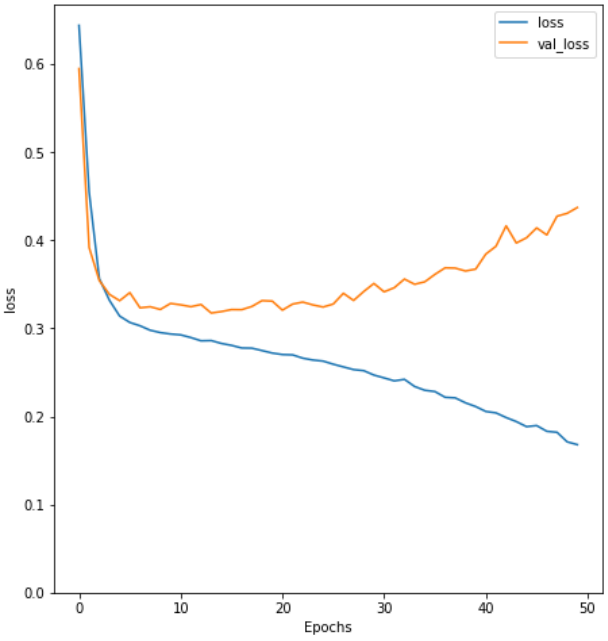
Fig(21). Measures of accuracy and val_accuracy on the training set and the validation set on 20 epochs.



Fig(22). Measures of Loss and val loss on the training set and the validation set on 20 epochs.



Fig(23). Measures of accuracy and val accuracy on the training set and the validation set on 50 epochs.

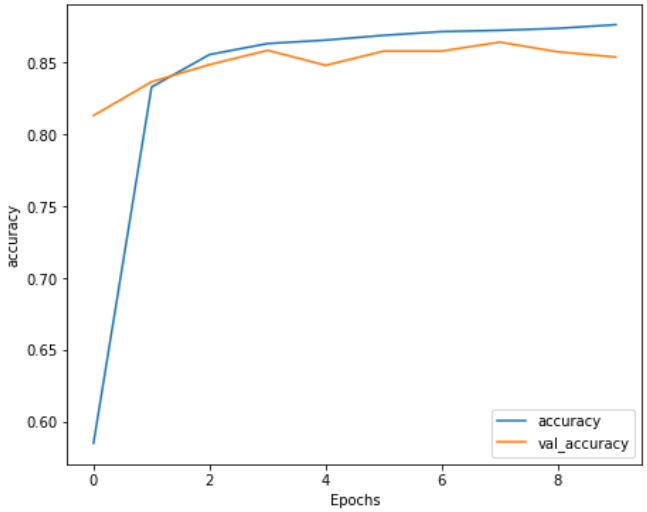


Fig(24). Measures of Loss and val loss on the training set and the validation set on 50 epochs.

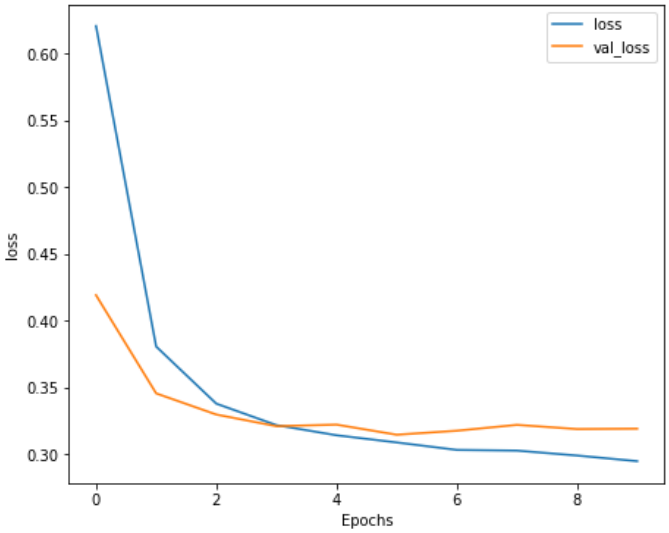
Table 9: Performance metrics on LSTM model

LSTM				
model.fit(input_train, y_train, epochs=10 , validation_split=0.2)				
	loss	Accuracy	val_loss	val_accuracy
1	0.6209	0.5852	0.4190	0.8130
2	0.3805	0.8328	0.3454	0.8365
3	0.3377	0.8554	0.3295	0.8484
4	0.3215	0.8630	0.3207	0.8583
5	0.3139	0.8654	0.3219	0.8479
6	0.3085	0.8688	0.3143	0.8578
7	0.3029	0.8714	0.3173	0.8578
8	0.3024	0.8722	0.3217	0.8641
9	0.2987	0.8736	0.3186	0.8573
10	0.2945	0.8762	0.3189	0.8536
model.fit(input_train, y_train, epochs=20, validation_split=0.2)				
11	0.2917	0.8776	0.3208	0.8609
12	0.2926	0.8752	0.3181	0.8578
13	0.2874	0.8783	0.3175	0.8542
14	0.2859	0.8802	0.3369	0.8635
15	0.2828	0.8802	0.3229	0.8547
16	0.2810	0.8820	0.3364	0.8396
17	0.2791	0.8817	0.3377	0.8620
18	0.2746	0.8826	0.3303	0.8479
19	0.2727	0.8850	0.3405	0.8583
20	0.2697	0.8874	0.3315	0.8521
model.fit(input_train, y_train, epochs=50, validation_split=0.2)				
21	0.2726	0.8831	0.3305	0.8552
22	0.2690	0.8828	0.3295	0.8547
23	0.2669	0.8848	0.3335	0.8531
24	0.2653	0.8849	0.3327	0.8443
25	0.2606	0.8871	0.3471	0.8578
26	0.2604	0.8866	0.3576	0.8333
27	0.2584	0.8886	0.3510	0.8547
28	0.2532	0.8898	0.3579	0.8438
29	0.2519	0.8894	0.3586	0.8542
30	0.2484	0.8923	0.3571	0.8542
31	0.2474	0.8926	0.3614	0.8365
32	0.2452	0.8930	0.3698	0.8536
33	0.2419	0.8959	0.3739	0.8474
34	0.2398	0.8959	0.3658	0.8240
35	0.2399	0.8958	0.3714	0.8417
36	0.2323	0.9000	0.3839	0.8323
37	0.2549	0.8897	0.3687	0.8328
38	0.2325	0.8992	0.3929	0.8427
39	0.2261	0.9030	0.4119	0.8375
40	0.2244	0.9050	0.4010	0.8526
41	0.2195	0.9048	0.4010	0.8484
42	0.2158	0.9090	0.4220	0.8479
43	0.2171	0.9074	0.4045	0.8359
44	0.2135	0.9083	0.4096	0.8359
45	0.2081	0.9109	0.4177	0.8370

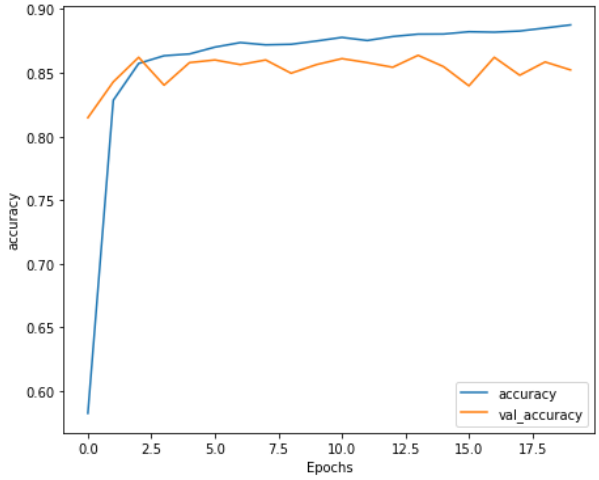
46	0.2051	0.9121	0.4313	0.8474
47	0.2006	0.9157	0.4329	0.8427
48	0.2020	0.9137	0.4252	0.8318
49	0.1950	0.9178	0.4492	0.8495
50	0.1909	0.9195	0.4574	0.8474



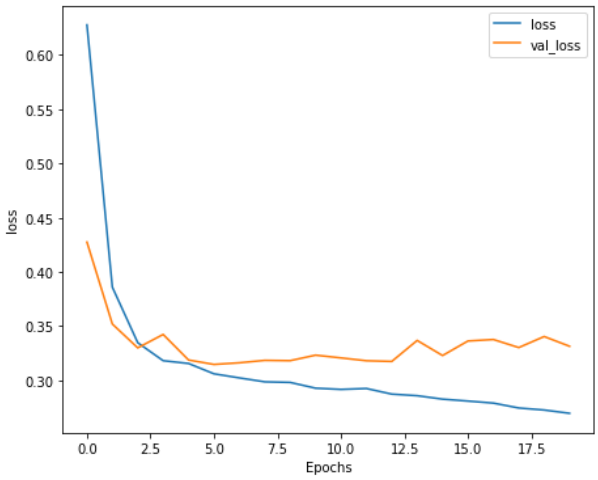
Fig(25). Measures of lstm accuracy and val _ accuracy on the training set and the validation set on 10epochs.



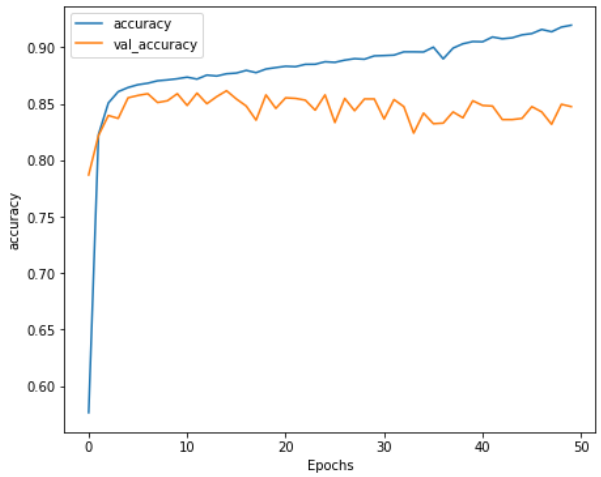
Fig(26). Measures of lstm loss and val _ loss on the training set and the validation set on 10epochs.



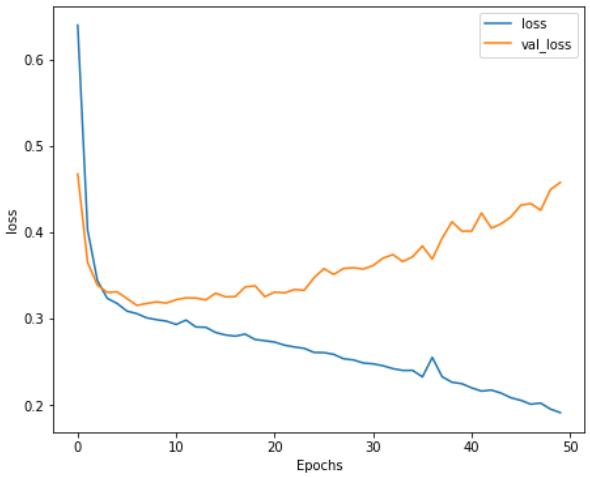
Fig(27). Measures of accuracy and val _ accuracy on the training set and the validation set on 20 epochs.



Fig(28). Measures of Loss and val _ loss on the training set and the validation set on 20 epochs.



Fig(29). Measures of accuracy and val _ accuracy on the training set and the validation set on 50 epochs.



Fig(30). Measures of Loss and val _ loss on the training set and the validation set on 50 epochs.

Table 10: Performance metrics on GRU model .

GRU				
model.fit(input_train, y_train, epochs=10, validation_split=0.2)				
	loss	**Accuracy**	**val_loss**	**val_accuracy**
1	0.6901	0.5613	0.6842	0.6112
2	0.6627	0.6739	0.6388	0.6700
3	0.5257	0.7699	0.4551	0.8004
4	0.3321	0.8581	0.3661	0.8396
5	0.2320	0.9097	0.3400	0.8566
6	0.1780	0.9376	0.3321	0.8708
7	0.1393	0.9559	0.3570	0.8702
8	0.1193	0.9632	0.3805	0.8730
9	0.0955	0.9730	0.4188	0.8672
10	0.0791	0.9790	0.4625	0.8652
model.fit(input_train, y_train, epochs=20, validation_split=0.2)				
11	0.0762	0.9782	0.4606	0.8622
12	0.0619	0.9843	0.4886	0.8588
13	0.0547	0.9862	0.5055	0.8588
14	0.0494	0.9874	0.5523	0.8506
15	0.0410	0.9905	0.6024	0.8568
16	0.0324	0.9934	0.6143	0.8530
17	0.0257	0.9954	0.6582	0.8502
18	0.0212	0.9970	0.6862	0.8480
19	0.0181	0.9976	0.7300	0.8494
20	0.0146	0.9982	0.7509	0.8478
model.fit(input_train, y_train, epochs=50, validation_split=0.2)				
21	0.0137	0.9983	0.7611	0.8542

22	0.0120	0.9985	0.7878	0.8520
23	0.0106	0.9987	0.8215	0.8516
24	0.0105	0.9984	0.7992	0.8486
25	0.0105	0.9984	0.7992	0.8486
26	0.0831	0.9714	0.6912	0.8354
27	0.0459	0.9844	0.7465	0.8470
28	0.0297	0.9908	0.7109	0.8486
29	0.0180	0.9959	0.7616	0.8508
30	0.0123	0.9980	0.7813	0.8496
31	0.0102	0.9986	0.8257	0.8484
32	0.0079	0.9991	0.8477	0.8442
33	0.0066	0.9993	0.8874	0.8452
34	0.0056	0.9994	0.9330	0.8442
35	0.0047	0.9995	0.9368	0.8446
36	0.0043	0.9996	0.9696	0.8440
37	0.0039	0.9996	0.9809	0.8436
38	0.0034	0.9997	1.0039	0.8434
39	0.0032	0.9997	1.0188	0.8438
40	0.0030	0.9997	1.0321	0.8438
41	0.0027	0.9997	1.0507	0.8442
42	0.0023	0.9998	1.0590	0.8440
43	0.0021	0.9998	1.0821	0.8440
44	0.0021	0.9998	1.1214	0.8470
45	0.0299	0.9916	1.0466	0.8426
46	0.0323	0.9895	0.9932	0.8478
47	0.0289	0.9906	0.8829	0.8486
48	0.0132	0.9962	0.9050	0.8506
49	0.0078	0.9983	0.9333	0.8472
50	0.0049	0.9992	0.9554	0.8478

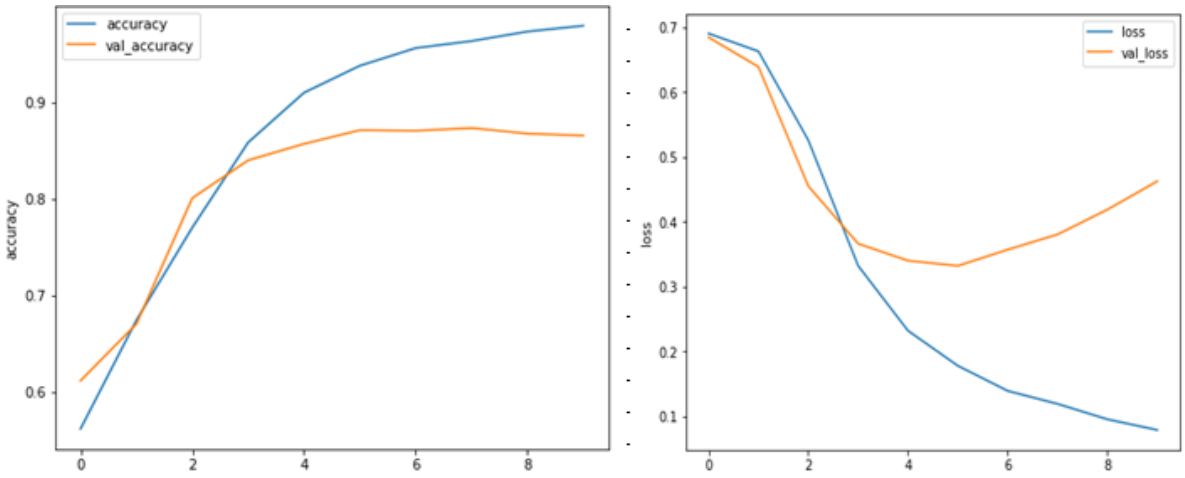


Fig (31). Measures of accuracy and val _ accuracy on the training set and the validation set on 10 epochs.

Fig (32). Measures of accuracy and val _ accuracy on the training set and the validation set on 10 epochs.

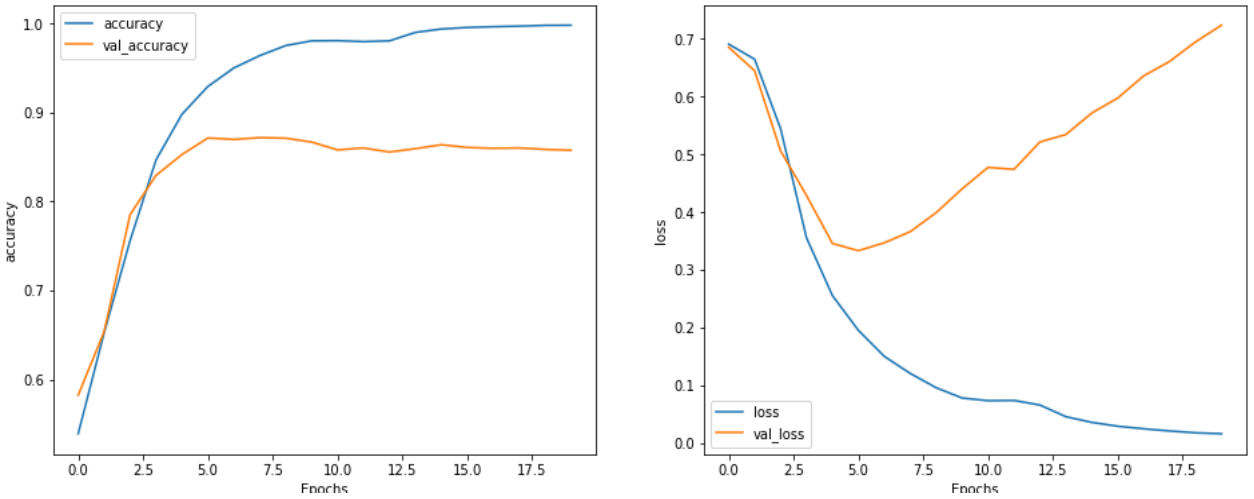


Fig (33). Measures of accuracy and val _ accuracy on the training set and the validation set on 20 epochs.

Fig (34). Measures of Loss and val _ loss on the training set and the validation set on 20epochs.

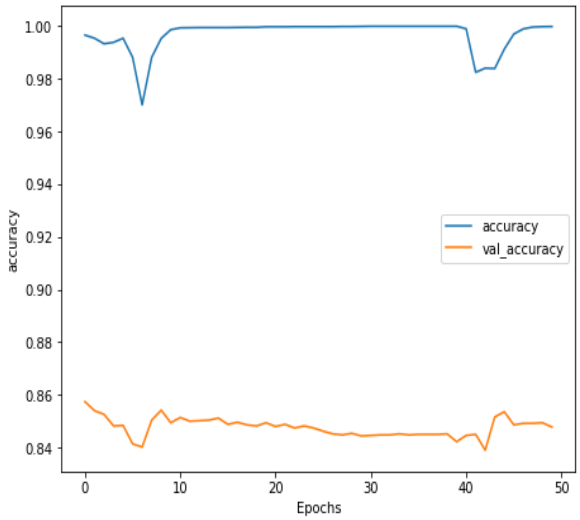
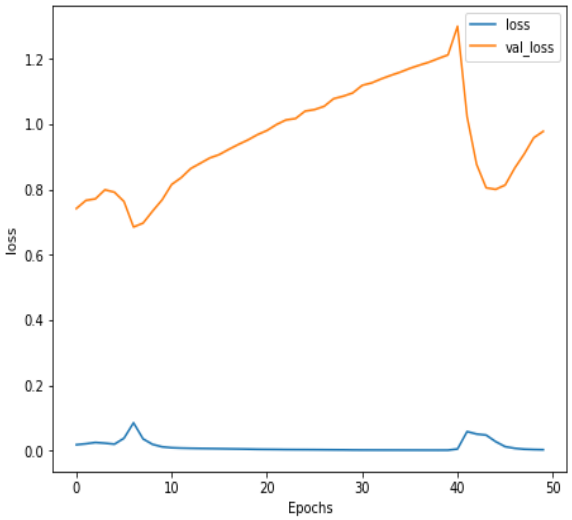


Fig (35). Measures of accuracy and val _ accuracy on the training set and the validation set on 50 epochs.



Fig(36). Measures of Loss and val _ loss on the training set and the validation set on 50 epochs.

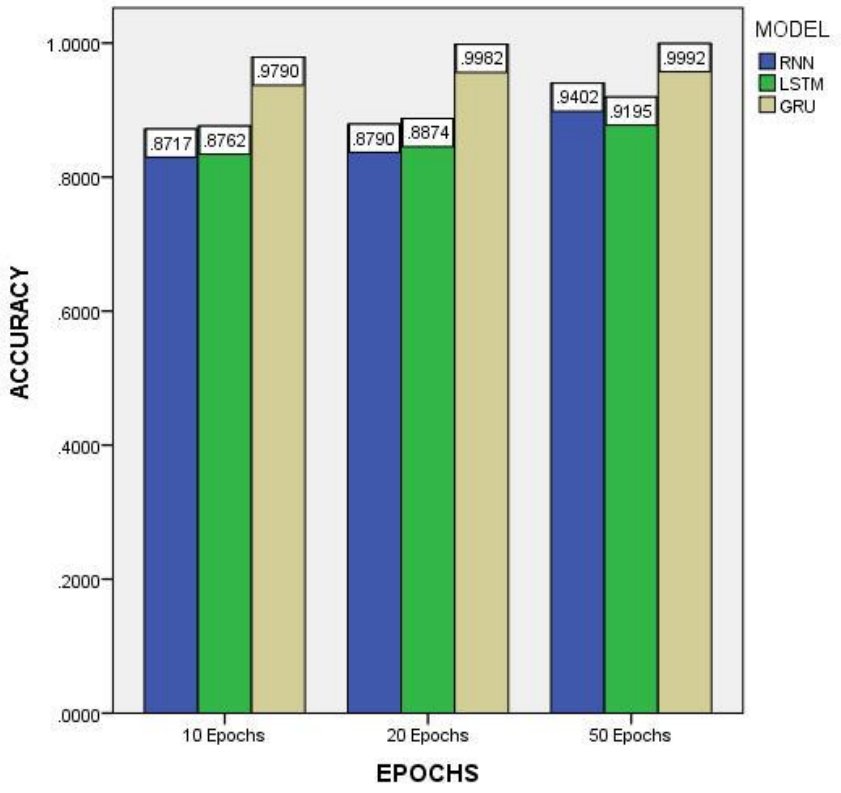


Fig (37). RRN – LSTM - GRU Accuracy Results

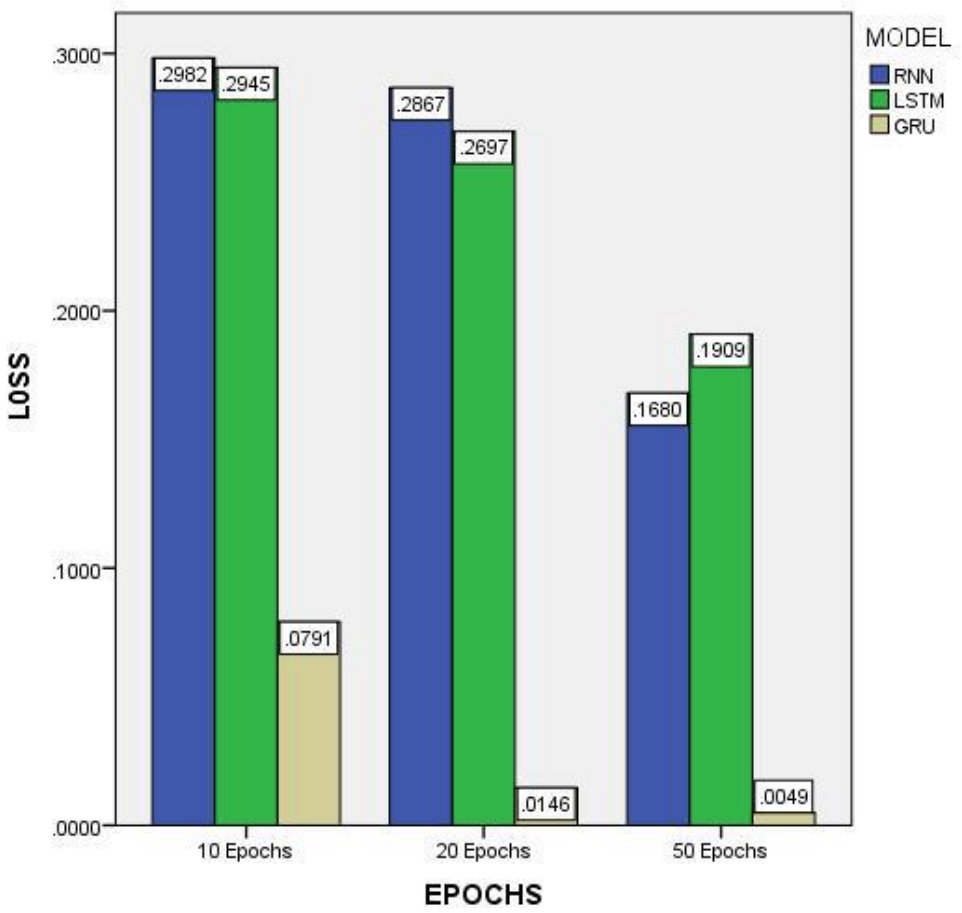


Fig (38). RRN – LSTM - GRU loss Results

Conclusion and Future Work

Part 1 . The result of the proposed edit based model that is applied on some questions with 60 students it's in details on table. The standard Hamming model achieved 33.3% correct answer with 0 correct defined distance value, and achieved 58.3% wrong answer with greater than or equa1 to 1 wrong defined distance value, and achieved 8.3% wrong answer with undefined distance value. But, The proposed edit based model achieved 75% correct answer with 0 correct defined distance value, and achieved 25% wrong answer with greater than or equa1 to 1 correct defined distance value. The future work will investigate how to integrate the semantic approach to achieve semantic similarity.

Part 2.From the above results, we have been differentiated among the students with their ignored or hidden knowledge. Using the edit-based algorithms such as Hamming, Levenshtein and Jaro_Winkler, the proposed HQAS has been achieved the high knowledge assessment compared withNthe low knowledge assessment in the TQAS where the big amount of hidden knowledge

was considered as non-significant knowledge. We have been introduced additional way with some improvements by the proposed edit-based HQAS to consider the hidden knowledge during the misspelling at the middle of the dual-ordered incomplete answer, during the misspelling at middle and the end of the intra-ordered incomplete answer, and during the misspelling due to switching in words of the intra-ordered complete answer. Also, we have implemented a proposed traversing by Breadth First Search (BFS) within the m-ary HQAS tree to implement the areas of a topic as depths and which of these depths have been significant knowledge.

Part 3. This part gave us a great start into the field of Neural networks. We obtained good learning with implementation of neural networks using Tensor Flow and keras and more libraries to support Arabic language. We were able to build an rrn , lstm , rgu neural network for predicts the correct answers after training , We got an accuracy on RRN upto 94% , LSTM upto 91% and GRU upto 99%. And on testing we got on RRN upto 85%, LSTM upto 84% and upto GRU 85%. In order to improve accuracy, we want to thoroughly analyse the current model and make additional modifications. To track changes in the system's performance, we'll put the stacked, bidirectional LSTM-based RNN a network into use.

Outline of Thesis

Chapter 1: Introduction

Chapter 2: Literature Survey

Chapter 3: Proposed Method

Chapter 4: Experimental Results and Discussion

Chapter 5: Conclusion and future work

References:

1. Ray, s., and Shaalan, K. (2016). A Review and Future Perspectives of Arabic Question Answering Systems. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3169-3190. DOI: 10.1109/TKDE.2016.2607201.

2. Kamaldeep, K., and Vishal, G. (2012). Name Entity Recognition for Punjabi Language ,*IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS)*, Vol. 2, No.3.

3. Rosso, P., Benajiba, Y., & Lyhyaoui, A. (2006). Towards an Arabic Question Answering system. *In: Proc. 4th Conf. on Scientific Research Outlook Technology Development in the Arab world*. SROIV. Damascus. Syria, pp. 11–14

4. poonam G., and Vishal G. Survey of Text Question Answering Techniques. (2012). *International Journal of Computer Applications* (0975 – 8887) Volume 53– No.4.

5. K. Ray, Santosh, and Shaalan, Khaled. "A Review and Future Perspectives of Arabic Question Answering Systems" IEEE Transactions on Knowledge and Data Engineering 28 (2016): 3169 - 3190.

6. Bakari, W., Bellot, P., & Neji, M. (2015). Literature Review of Arabic Question-Answering: Modeling, Generation, Experimentation and Performance Analysis. In Flexible Query Answering Systems 2015 (pp. 321-334).

7. Syahidah Sufi Haris and Nazliamar, Bloom's Taxonomy Question Categorization Using Rules and NGram Approach, Journal of Theoretical and Applied Information Technology Vol.76. No.3, 30th June 2015.

8. Poonguzhali, K Lakshmi, "Temporal Question Answering System: A Survey", JETIR June, Volume 6, Issue 6, ISSN 2349-5162, 2019.

9. K.S.D. Ishwari, A.K.R.R.Aneeze, S.Sudheesan, H.J.D.A. Karunaratne, A. Nugaliyadde, Y.Mallawarrachchi, "Advances in Natural Language Question Answering: A Review", arXiv: 1904.05276 [cs.CL], Apr 2019.

10. Wen-Chih. Chang and Ming-Shun. Chung, "Automatic Applying Bloom's Taxonomy to Classify and Analysis the Cognition Level of English Question Items", Joint Conferences on Pervasive Computing (JCPC). December 2009, pp. 727-734, 2009.

11. Asmita Dhokrat ,Gite Hanumant R and C.Namrata Mahender, “Automated Answering for subjective Examination “, International Journal of Computer Applications (0975 – 8887) Volume 56– No.14, October 2012.

12. Naveed Azim,Imran Naqvi and Dr. Kashif ur Rehman "Online Examination System and Assessment of Subjective Expression" International Conference on Education Technology and Computer 2009.
13. Kepei Zhang and Jieyu Zhao “ A Chinese Question-Answering System with Question Classification and Answer Clustering “, Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010) .

14. Volery, T and Lord “ Critical success factors in online education “, The International Journal of Educational Management ,2MCB University Press[ISSN 0951-354X] ,2000.

15. Hyo-Jung Oh, Chung-Hee Lee, Hyeon-Jin Kim, Myung-Gil Jang” Descriptive Question Answering in Encyclopedia “,Proceedings of the ACL Interactive Poster and Demonstration Sessions,pages 21–24, Ann Arbor, June 2005. c2005 Association for Computational Linguistics .

16. Waheeb Ahmed,Ajusha Pv and Babu Anto P,” Web-Based Arabic Question Answering System using Machine Learning Approach”, Volume 8, No. 1, Jan-Feb 2017 International Journal of Advanced Research in Computer Science.

17. Garima Nanda ,Mohit Dua And Krishma Singla” A Hindi Question Answering System using Machine Learning Approach “, 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT) .

18. Antoun W, Baly F, Hajj H. 2020. AraBERT: transformer-based model for Arabic languageunderstanding. In: Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection. Marseille: European Language Resource Association, 9–15.

19. Almiman A, Osman N, Torki M. 2020. Deep neural network approach for Arabic community question answering. Alexandria Engineering Journal 59(6):4427–4434DOI 10.1016/j.aej.2020.07.048.

20. Karpukhin V, Oguz B, Min S, Lewis P, Wu L, Edunov S, Chen D, Yih WT. 2020. Dense passageretrieval for open-domain question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 6769–6781.

21. Guu K, Lee K, Tung Z, Pasupat P, Chang MW. 2020. Realm: retrieval-augmented language model pre-training. arXiv preprint. Available at arXiv:2002.08909.

22. Huang Z, Xu S, Hu M, Wang X, Qiu J, Fu Y, Zhao Y, Peng Y, Wang C. 2020. Recent trends in deep learning based open-domain textual question answering systems. IEEE. Access 8:94341–94356 DOI 10.1109/ACCESS.2020.2988903.

23. Vinyals, O. , and Le, Q. , "A neural conversational model,"arXiv preprint, arXiv:1506.05869, 2015.

Publications:

1. Samah AliAl-azani[a]C.Namrata Mahender[a]MohammedHasan[b] "Development of an Arabic HQAS-based ASAG to consider an ignored knowledge in misspelled multiple words short answers "Global Transitions Proceedings, https://doi.org/10.1016/j.gltp.06.02.2022. **(Elsevier)**

2. Samah Ali Alazani, C. Namrata Mahender ", Improve hamming character difference based-on derivative lexical similarity and right space padding ", DOI: https://doi.org/10.36909/jer.ICETET.14979 , "Journal of Engineering Research"- , October 2021. **(Scopus)**

3. Samah Ali Alazani, C. Namrata Mahender, "Subjective questions answering system concepts: Approaches and tracks". the Seybold report journal ,DOI 10.5281/zenodo.7003418 -2022. **(Scopus)**

4. Samah Ali Alazani, C. Namrata Mahender, "Rule Based Part of Speech Tagger for Arabic Question Answering System ",International Conference on Communication, Computing and Electronics Systems pp 733–743 , , https://link.springer.com/chapter/10.1007/978-981-33-4909-4_58 , March 2021.(**(Scopus)**

5. Samah Ali Alazani, C. Namrata Mahender ,"Evaluating candidate Answers based on derivative lexical similarity and space padding for the Arabic language ",International Conference on Computational Intelligence in Data Science ,ICCIDS 2021: Computational Intelligence in Data Science pp 102–112 , https://link.springer.com/chapter/10.1007/978-3-030-92600-7_10, December 2021. **(Scopus)**

6. Samah Ali Alazani, C. Namrata Mahender ,"Rule Based Question Generation for Arabic Text: Question Answering System **",**DSMLAI '21': Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence ,Pages 7–12 , https://doi.org/10.1145/3484824.3484882 , August 2021.**(ACM)**

7. Samah Ali Alazani, C. Namrata Mahender ," Presentation of an efficient Top-Down parsing to check grammatically the syntax in Arabic verbal and nominal sentences, 4th International Conference on Recent Trends in Engineering and Technology2022 .